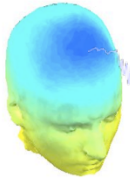
STUDY clustering overview



STEP 1

Build a STUDY

STEP 2

Precompute the data

STEP 3

Precluster the data

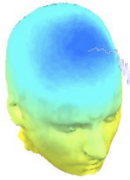
STEP 4

Cluster the data

STEP 5

Edit/view the clusters

STUDY clustering overview



STEP 1

Build a STUDY

STEP 2

Precompute the data

STEP 3

Precluster the data

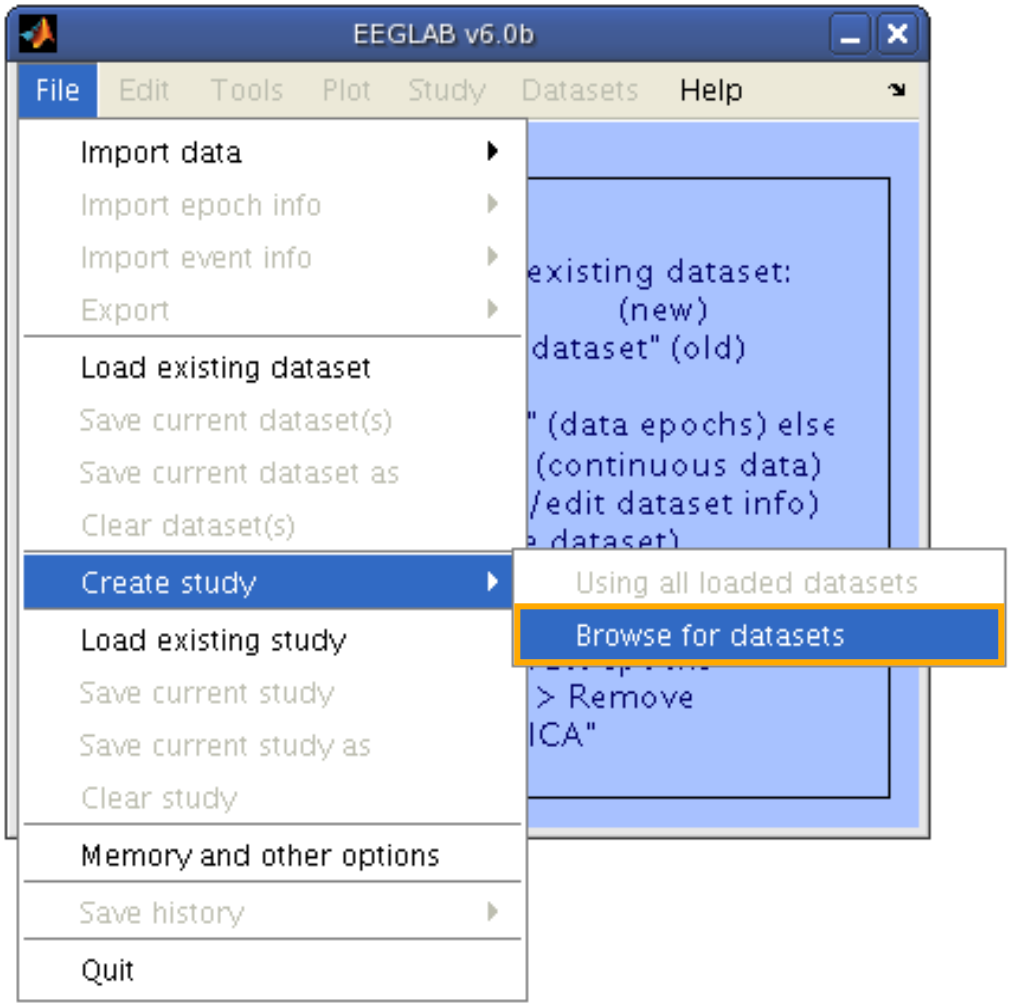
STEP 4

Cluster the data

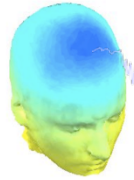
STEP 5

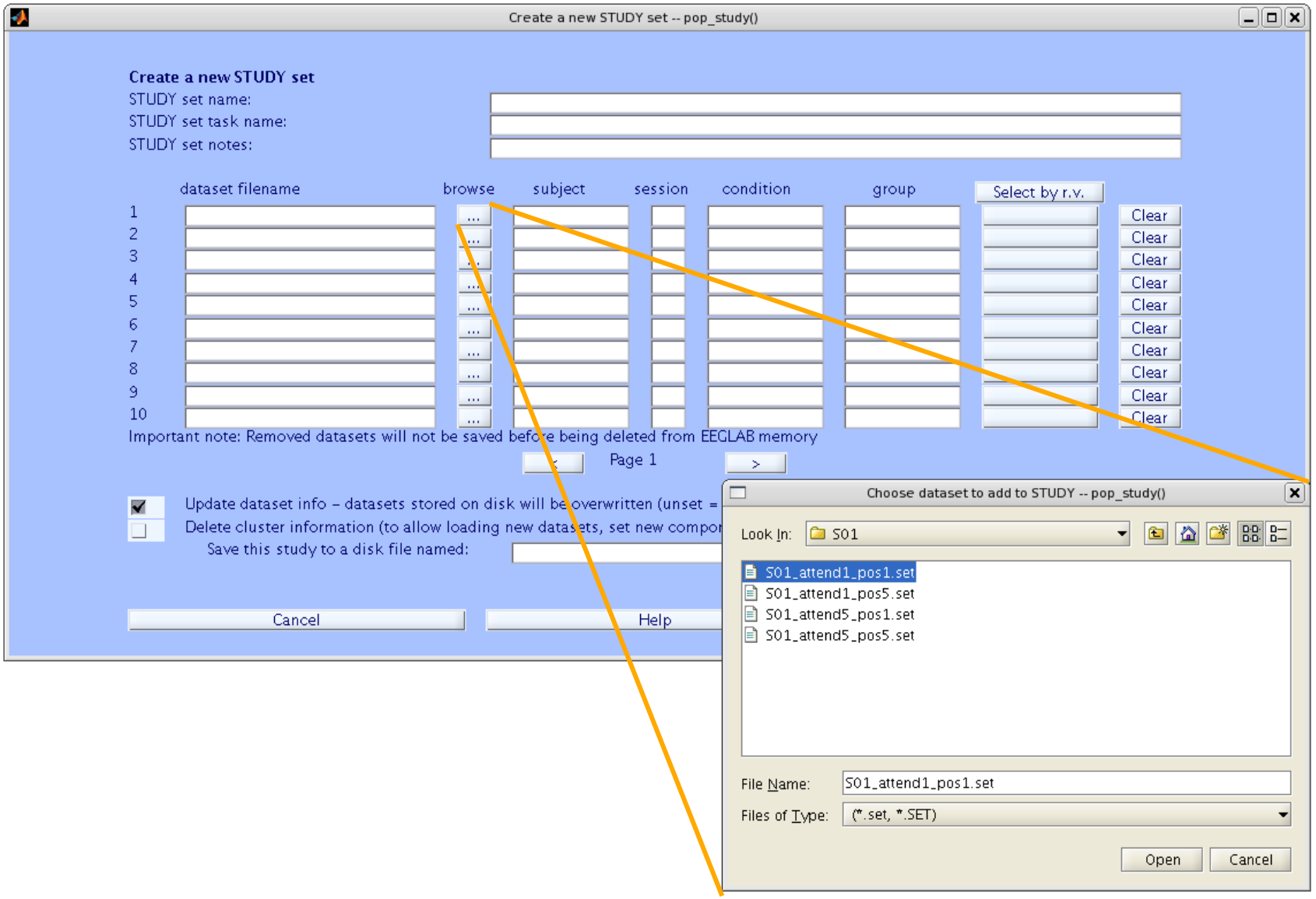
Edit/view the clusters

Build a STUDY

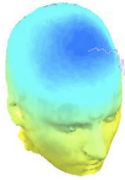


Build a STUDY, cont'd





Edit dataset info



pop_study(): Pre-select components

Enter maximum residual (topo map – dipole proj.) var. (in %)
NOTE: This will delete any existing component clusters!

15

Keep only in-brain dipoles.

Cancel | Help | Ok

Create a new STUDY set -- pop_study()

Edit STUDY set information

STUDY set name: Attention

STUDY set task name: 5-box

STUDY set notes:

	dataset filename	browse	subject	session	condition	group	Select by r.v.	
1	IDY/S01/S01_attend1_pos1.set	...	S01	1	TargAttnL	normals	All comp.	Clear
2	IDY/S01/S01_attend1_pos5.set	...	S01	1	NONTargAttnL	normals	All comp.	Clear
3	IDY/S01/S01_attend5_pos5.set	...	S01	1	TargAttnR	normals	All comp.	Clear
4	IDY/S01/S01_attend5_pos1.set	...	S01	1	NONTargAttnR	normals	All comp.	Clear
5	IDY/S02/S02_attend1_pos1.set	...	S02	1	TargAttnL	normals	All comp.	Clear
6	IDY/S02/S02_attend1_pos5.set	...	S02	1	NONTargAttnL	normals	All comp.	Clear
7	IDY/S02/S02_attend5_pos5.set	...	S02	1	TargAttnR	normals	All comp.	Clear
8	IDY/S02/S02_attend5_pos1.set	...	S02	1	NONTargAttnR	normals	All comp.	Clear
9	IDY/S03/S03_attend1_pos1.set	...	S03	1	TargAttnL	normals	All comp.	Clear
10	IDY/S03/S03_attend1_pos5.set	...	S03	1	NONTargAttnL	normals	All comp.	Clear

Important note: Removed datasets will not be saved before being deleted from EEGLAB memory

< Page 1 >

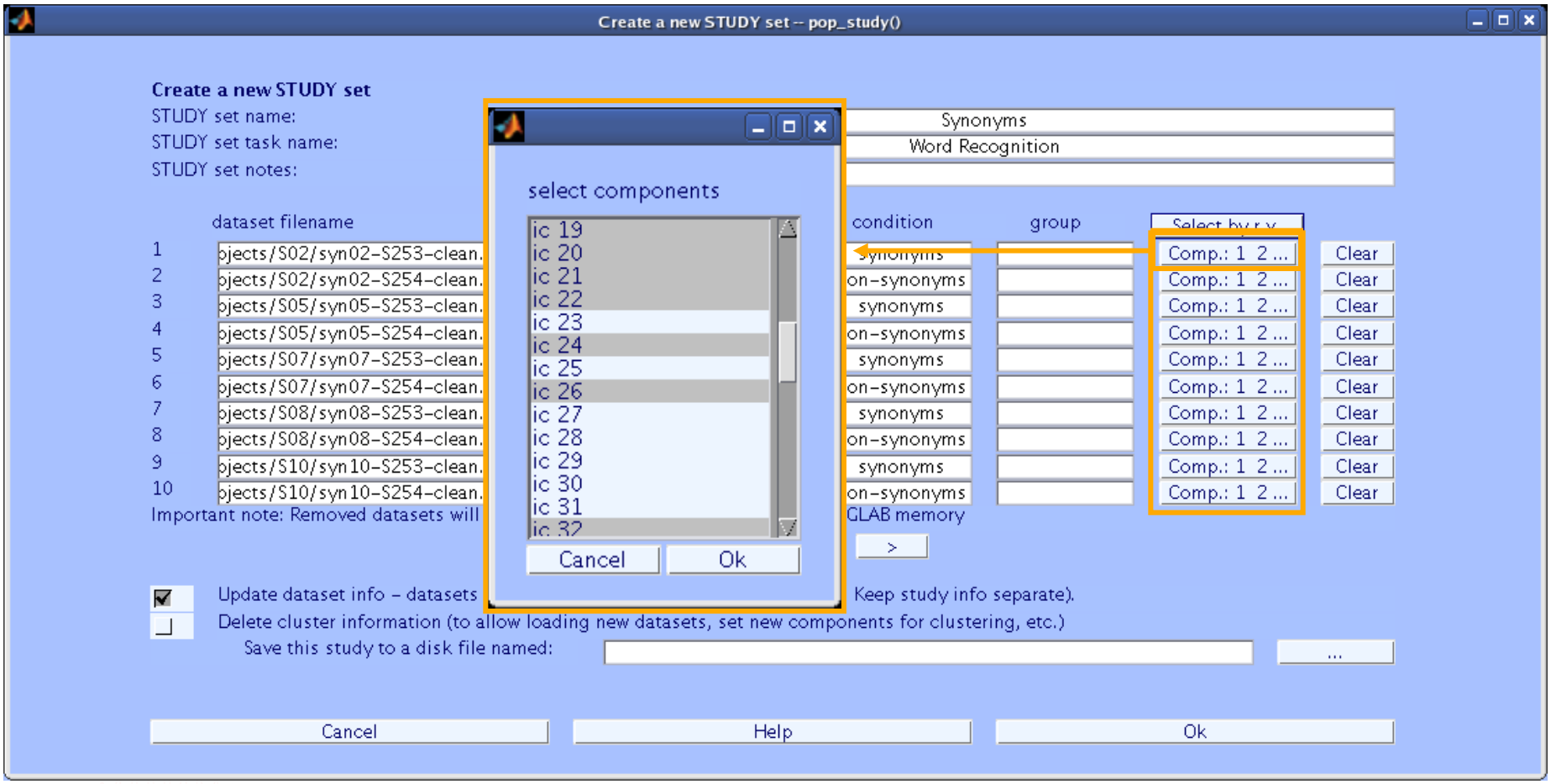
☑ Update dataset info – datasets stored on disk will be overwritten (unset = Keep study info separate).

☐ Delete cluster information (to allow loading new datasets, set new components for clustering, etc.)

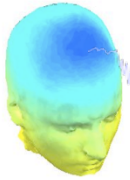
☐ Re-save STUDY. Uncheck and use menu File > Save study as to save under a new filename

Cancel | Help | Ok

ICs to cluster

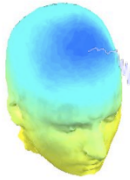


STUDY structure



```
STUDY =
            name: 'Synonyms'
            task: 'Word Recognition'
           notes: ''
        filename: 'workshop.study'
         cluster: [1x1 struct]
         history: [1x6654 char]
     datasetinfo: [1x10 struct]
        filepath: '/data/STUDY'
         subject: {'S02'  'S05'  'S07'  'S08'  'S10'}
           group: {}
         session: []
       condition: {'non-synonyms'  'synonyms'}
          setind: [2x5 double]
             etc: [1x1 struct]
        preclust: [1x1 struct]
           saved: 'no'
         changrp: []
>>
```

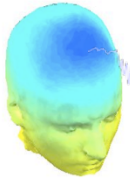
Subject info in STUDY structure



```
>> STUDY.datasetinfo

ans =

1x10 struct array with fields:
    filepath
    filename
    subject
    session
    condition
    group
    comps
    index

>>
```

Gives information for each subject

STUDY clustering overview



STEP 1

Build a STUDY

STEP 2

Precompute the data

STEP 3

Precluster the data

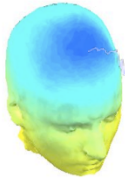
STEP 4

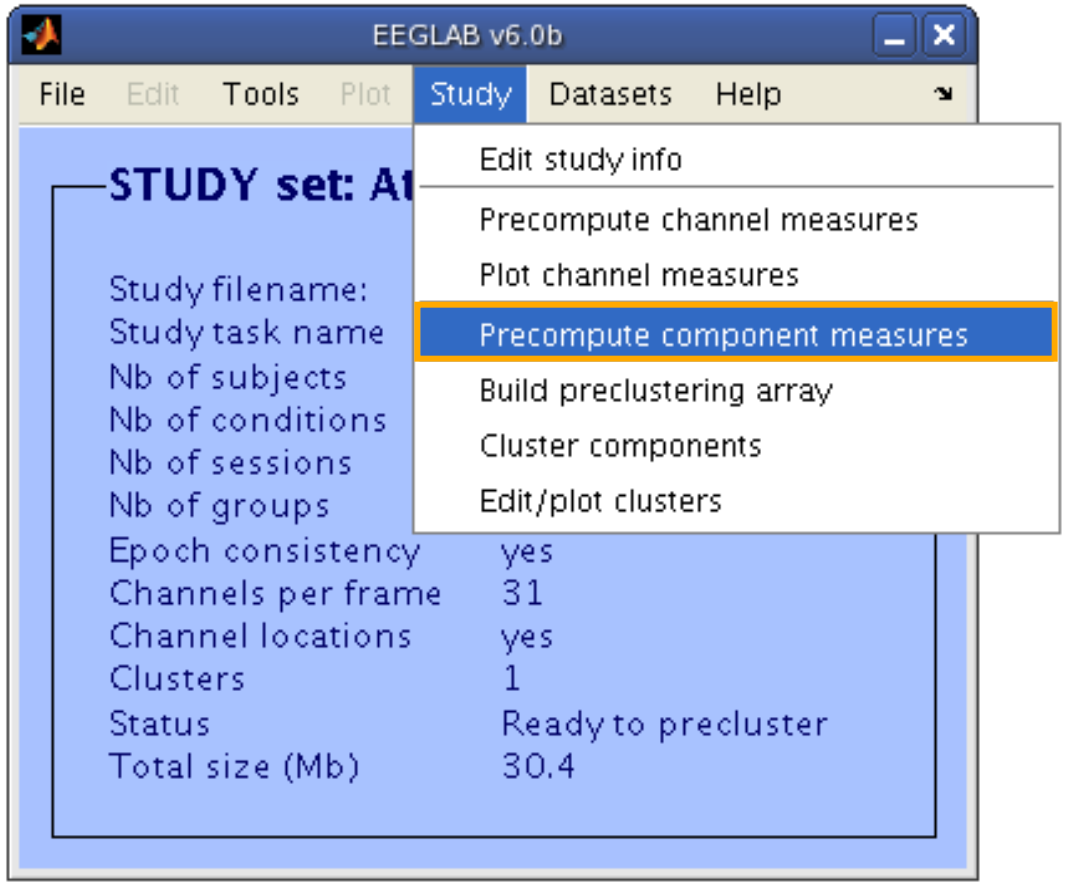
Cluster the data

STEP 5

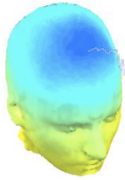
Edit/view the clusters

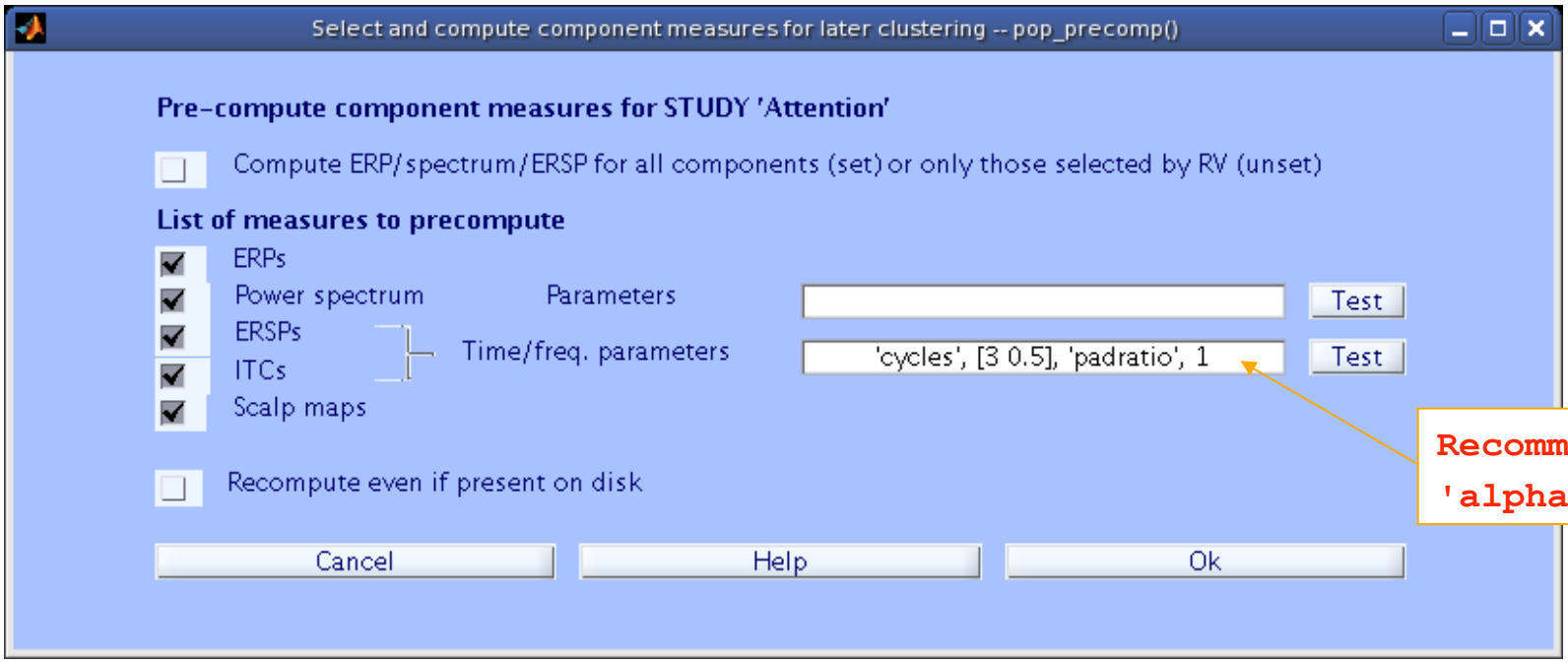
Precompute data measures





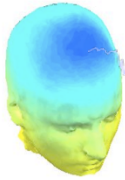
Precompute data measures





TIP: Compute all measures and so you can test different combinations for later clustering

STUDY clustering overview



STEP 1

Build a STUDY

STEP 2

Precompute the data

STEP 3

Precluster the data

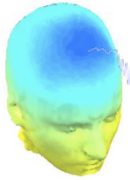
STEP 4

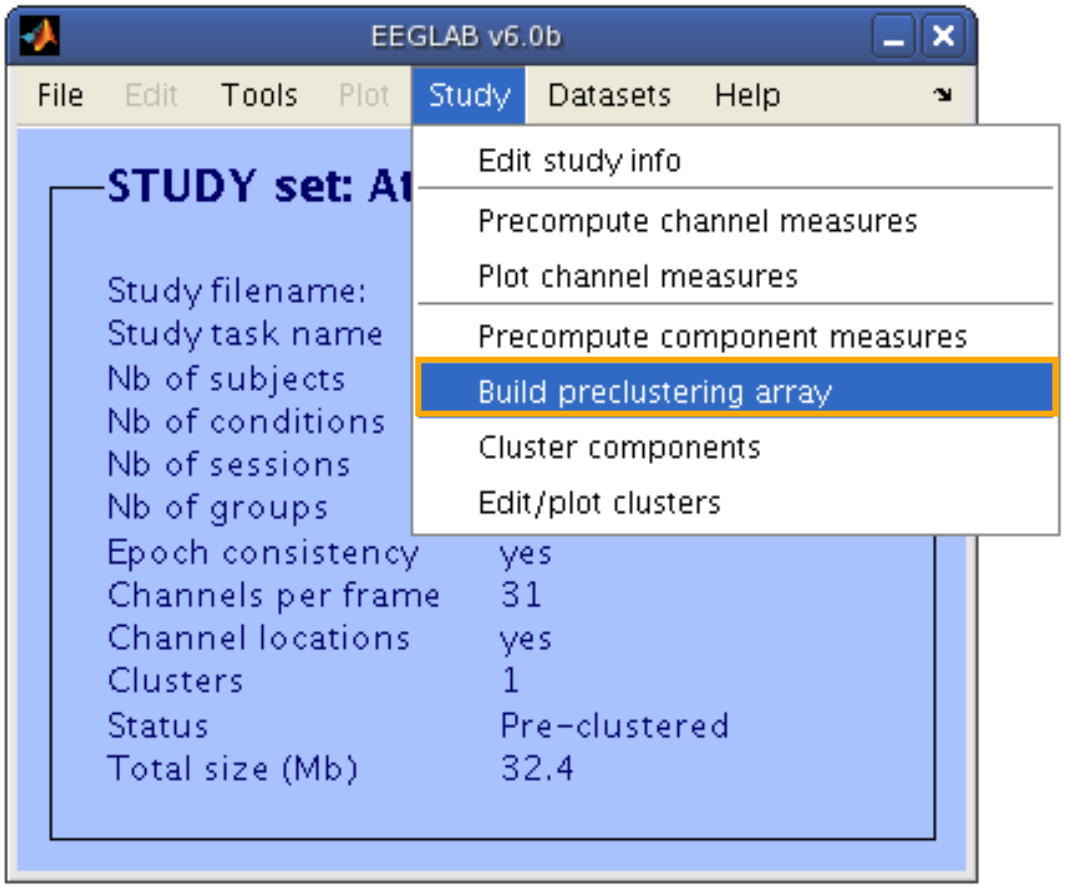
Cluster the data

STEP 5

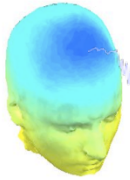
Edit/view the clusters

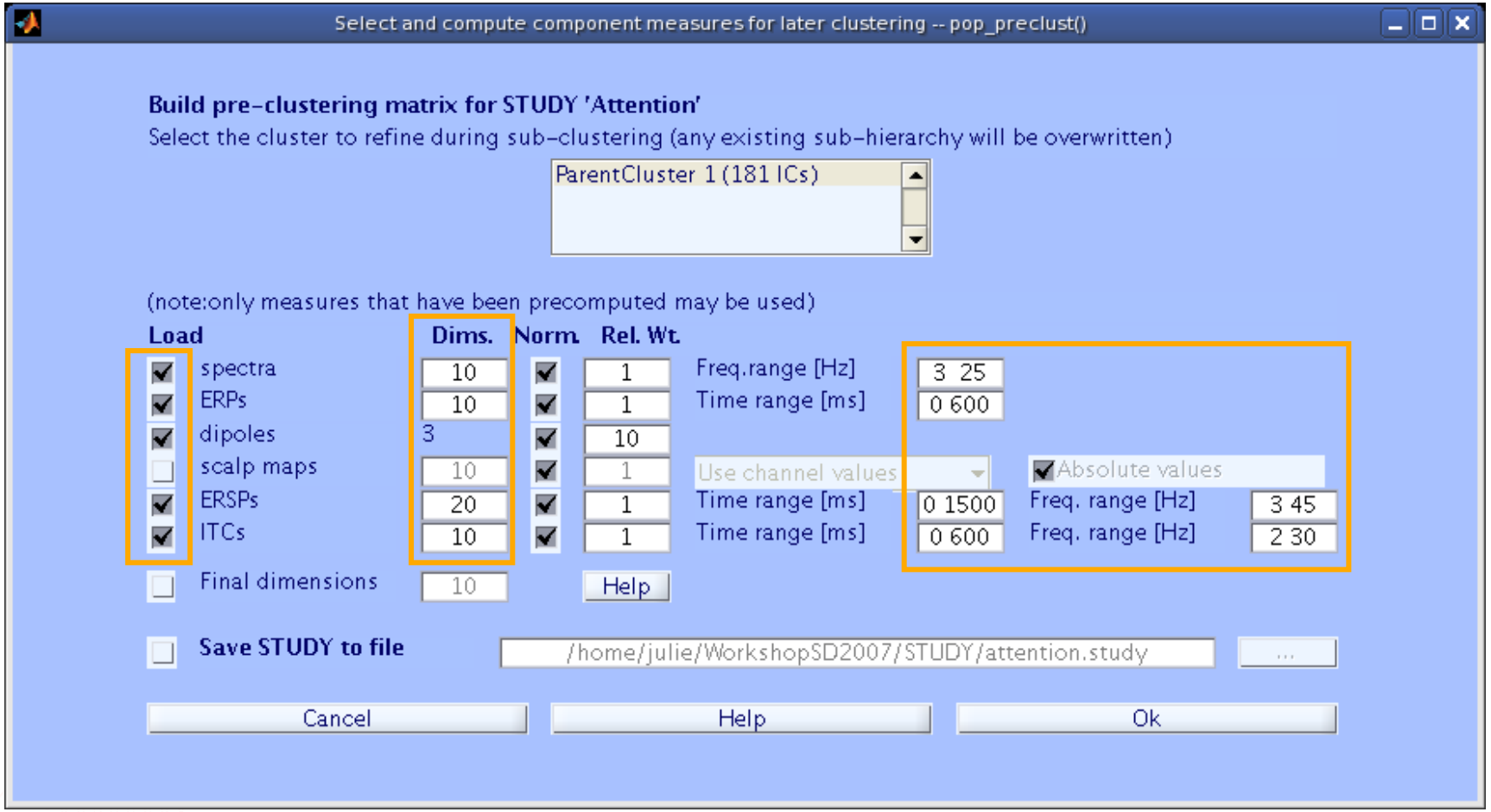
Precluster the data



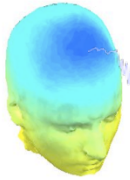


Precluster the data





STUDY clustering overview



STEP 1

Build a STUDY

STEP 2

Precompute the data

STEP 3

Precluster the data

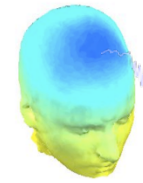
STEP 4

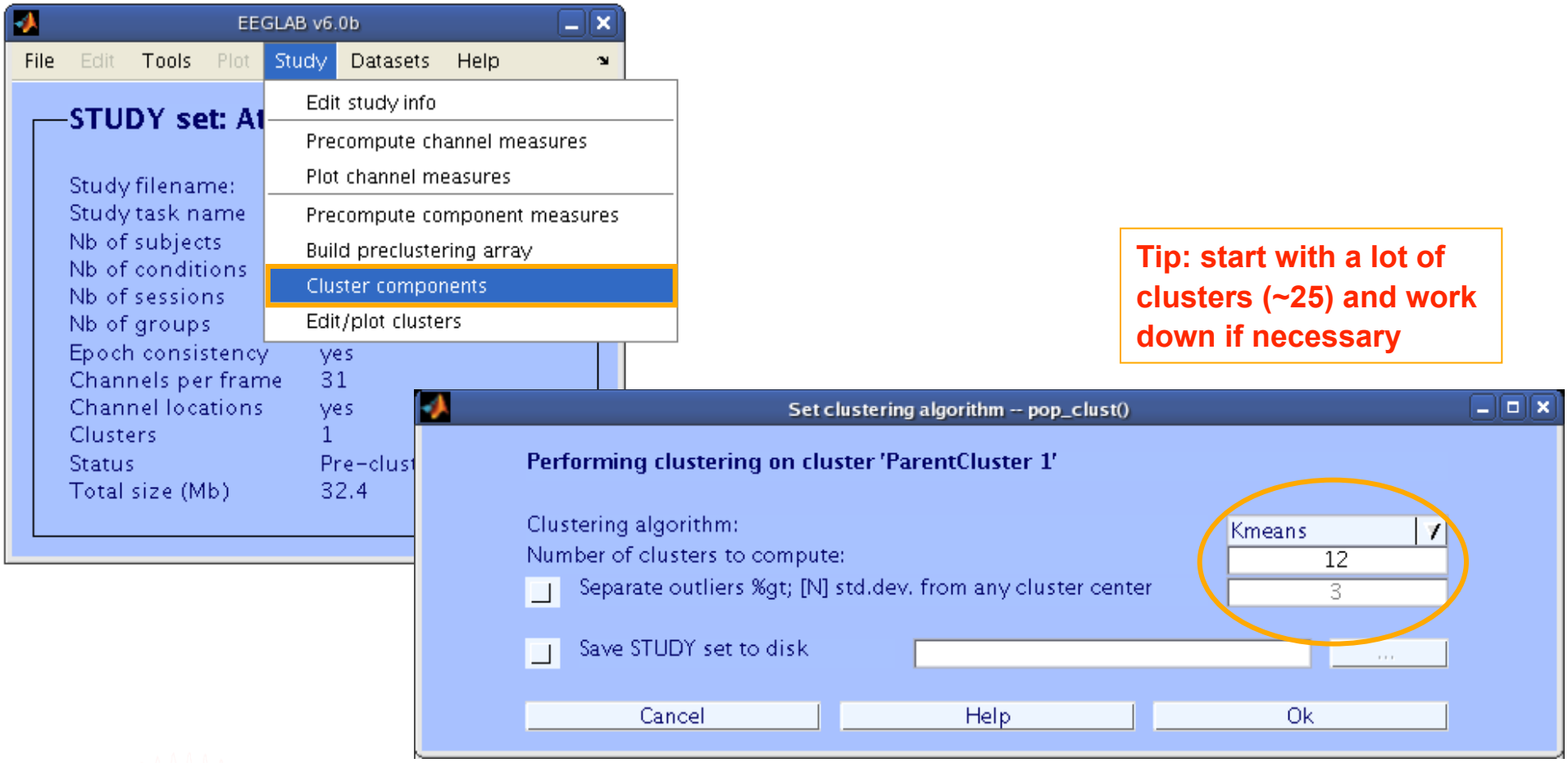
Cluster the data

STEP 5

Edit/view the clusters

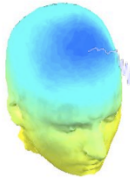
Cluster components





Tip: start with a lot of clusters (~25) and work down if necessary

STUDY clustering overview



STEP 1

Build a STUDY

STEP 2

Precompute the data

STEP 3

Precluster the data

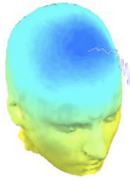
STEP 4

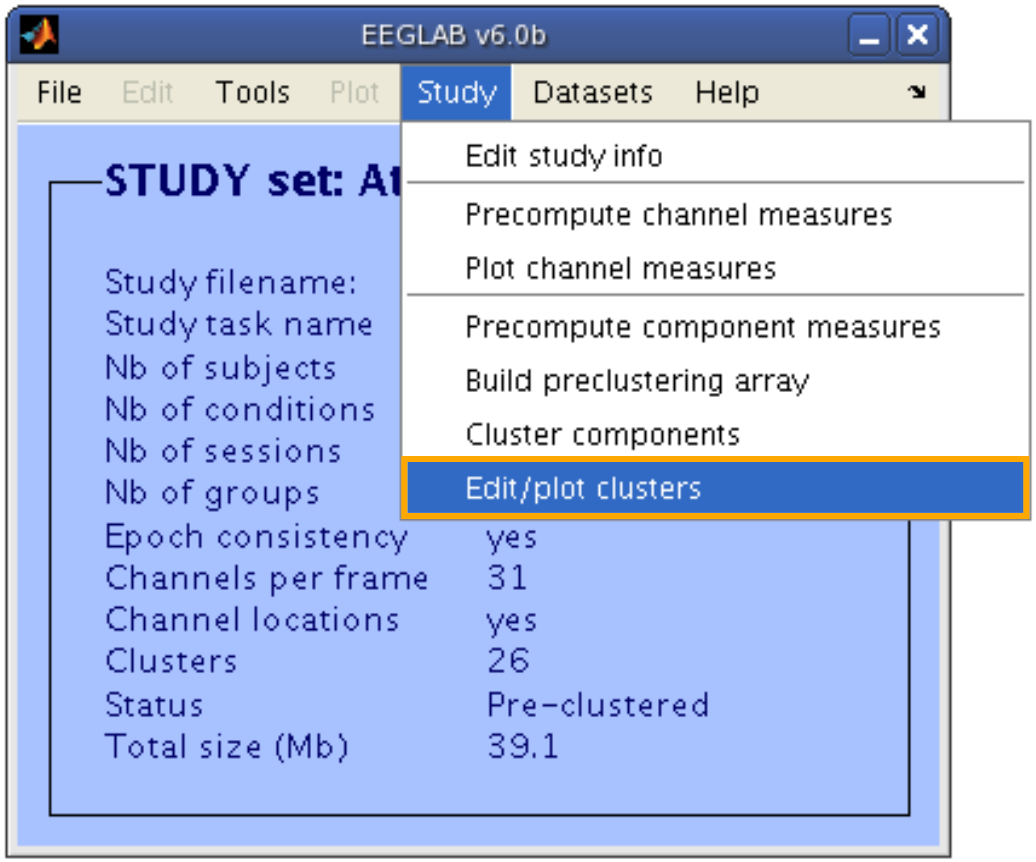
Cluster the data

STEP 5

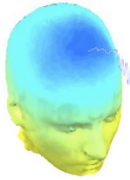
Edit/view the clusters

View and edit clusters

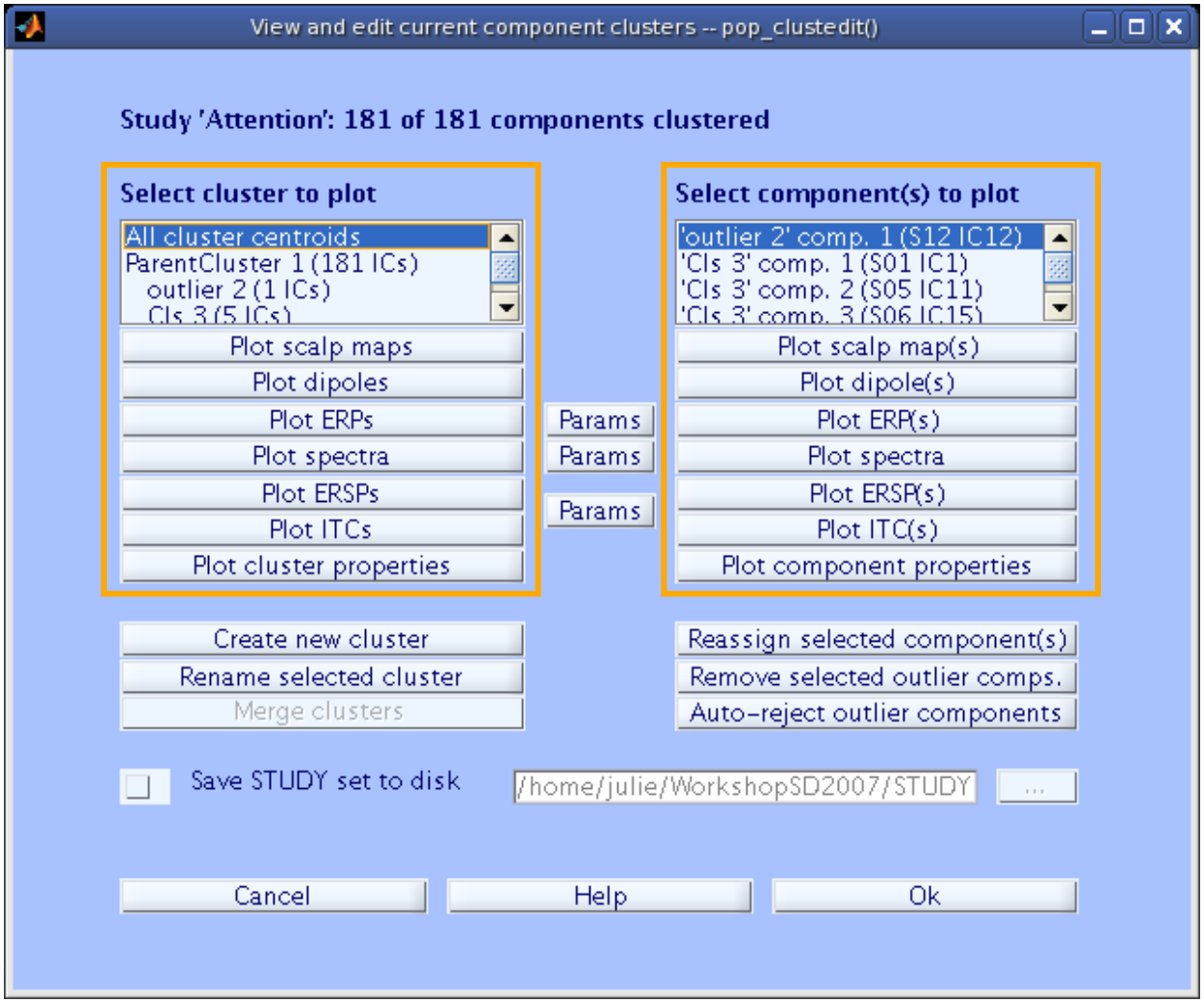




View and edit clusters

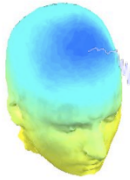


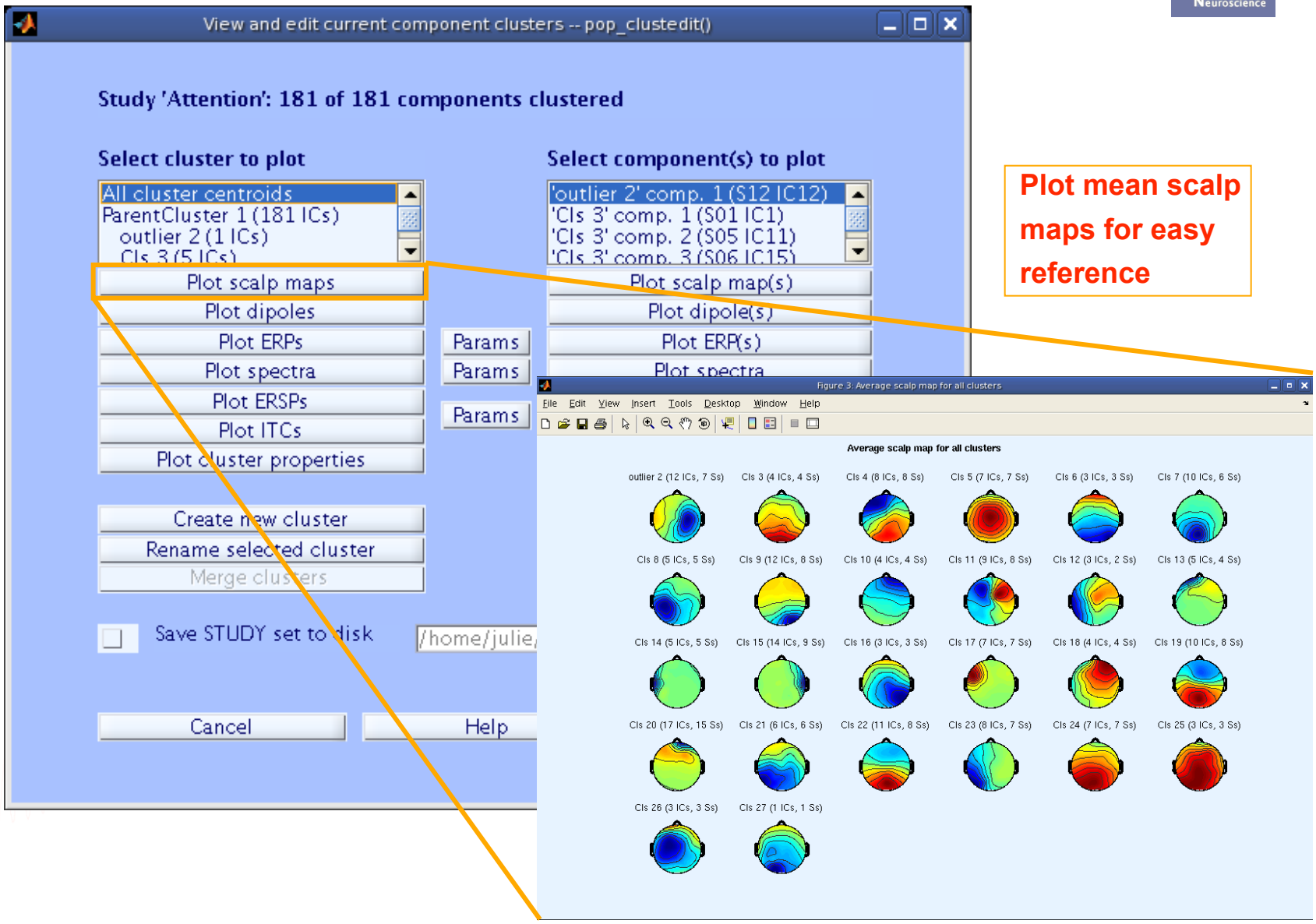
LEFT SIDE:
Plot measures of full clusters



RIGHT SIDE:
Plot measures of individual components

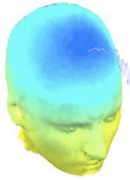
Plot cluster data

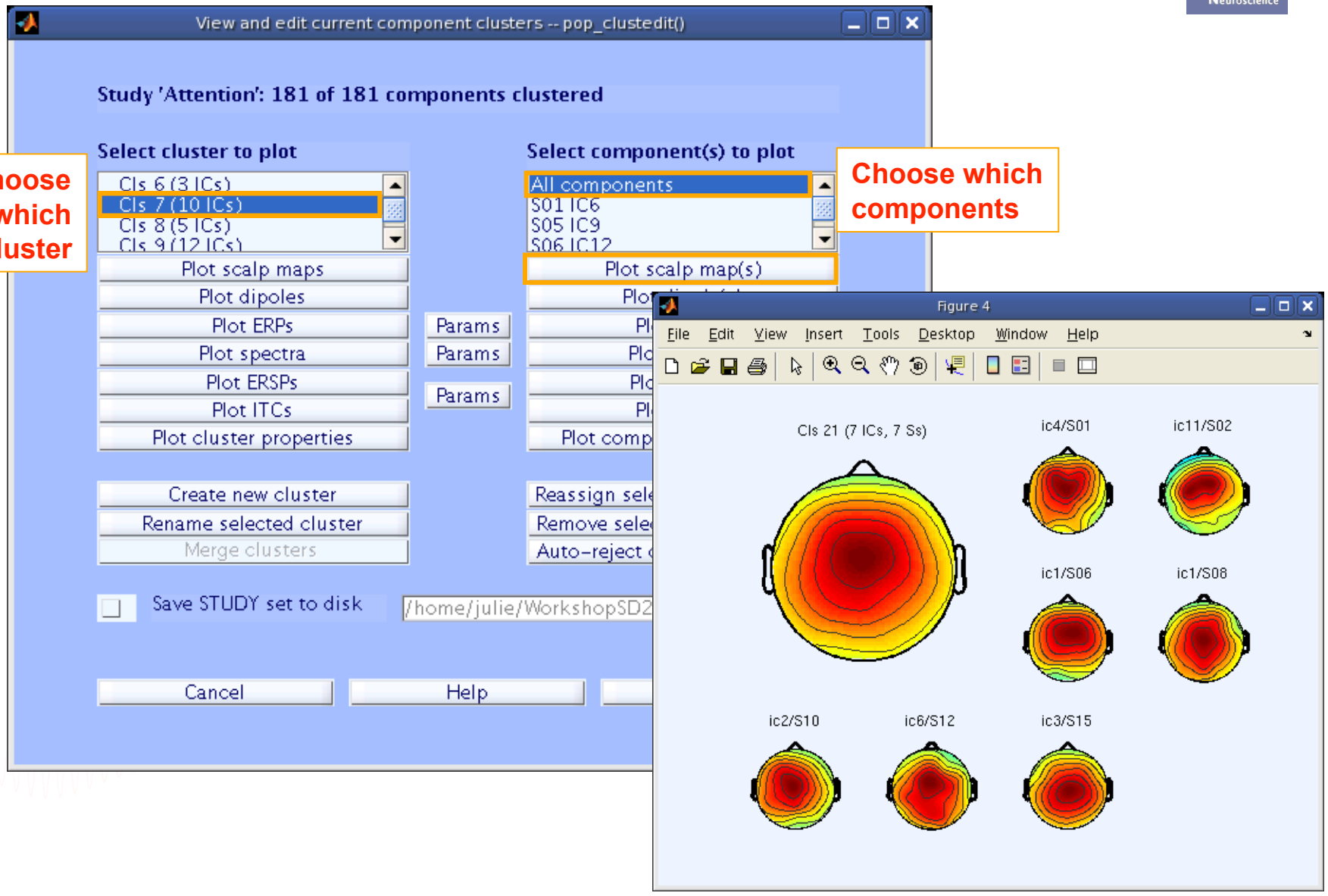




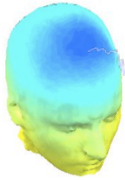
Plot mean scalp maps for easy reference

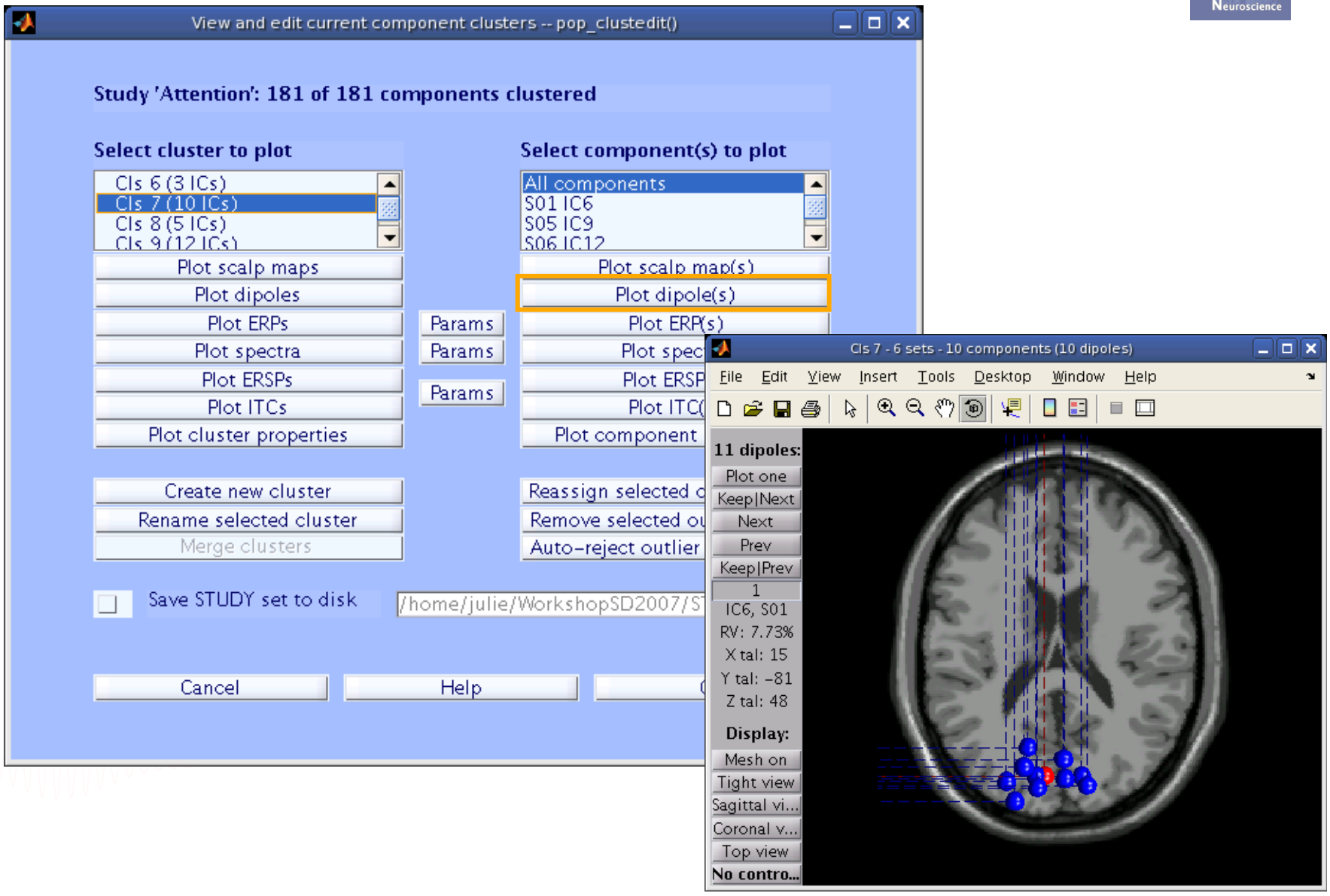
Plot cluster data



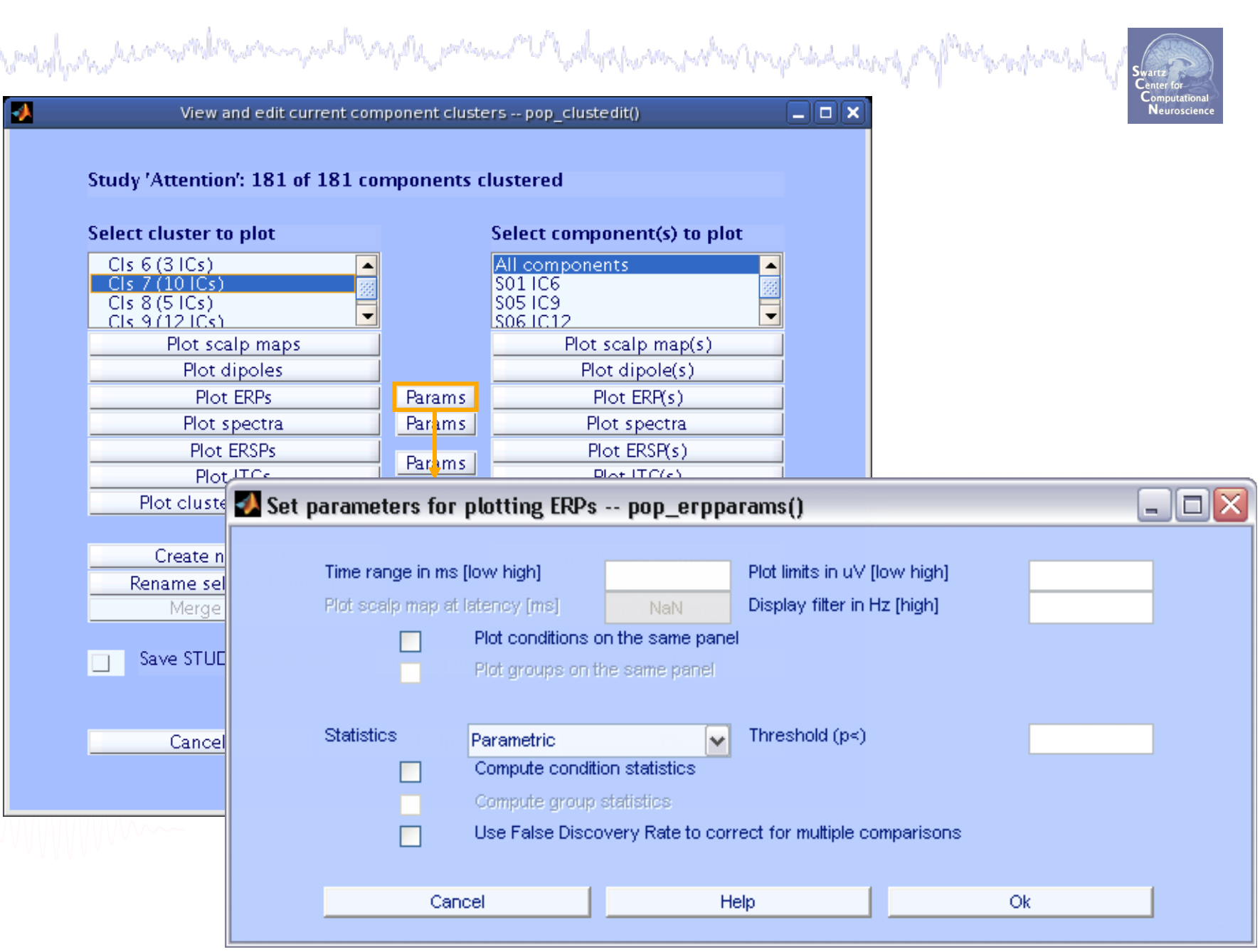


Plot clusters

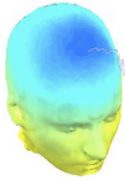




Plot ERPs

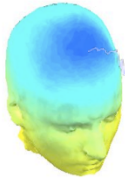

View and edit current component clusters -- pop_clustedit()
Study 'Attention': 181 of 181 components clustered
Select cluster to plot
Cls 6 (3 ICs)
Cls 7 (10 ICs)
Cls 8 (5 ICs)
Select component(s) to plot
All components
S01 IC6
S05 IC9
Plot scalp maps
Plot dipoles
Plot ERPs
Plot spectra
Plot ERSPs
Params
Plot scalp map(s)
Plot dipole(s)
Plot ERP(s)
Plot spectra
Plot ERSP(s)
Create n
Rename sel
Merge
Save STUD
Cancel
Set parameters for plotting ERPs -- pop_erppparams()
Time range in ms [low high]
Plot scalp map at latency [ms]
NaN
Plot limits in uV [low high]
Display filter in Hz [high]
Plot conditions on the same panel
Plot groups on the same panel
Statistics
Parametric
Threshold (p<)
Compute condition statistics
Compute group statistics
Use False Discovery Rate to correct for multiple comparisons
Cancel
Help
Ok

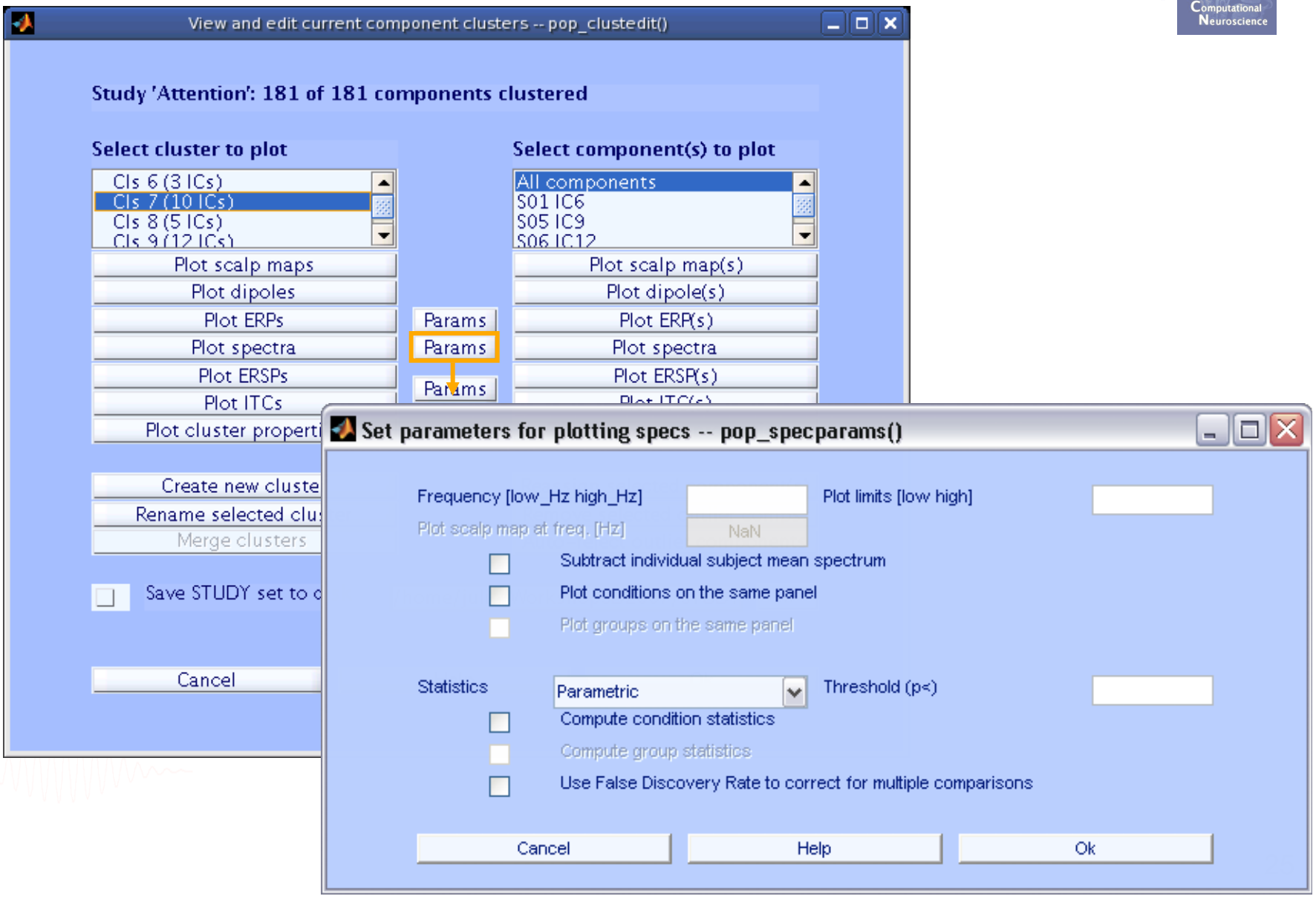
Plot cluster ERP



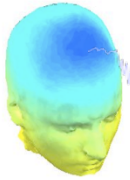
Each blue trace is the ERP of a different component

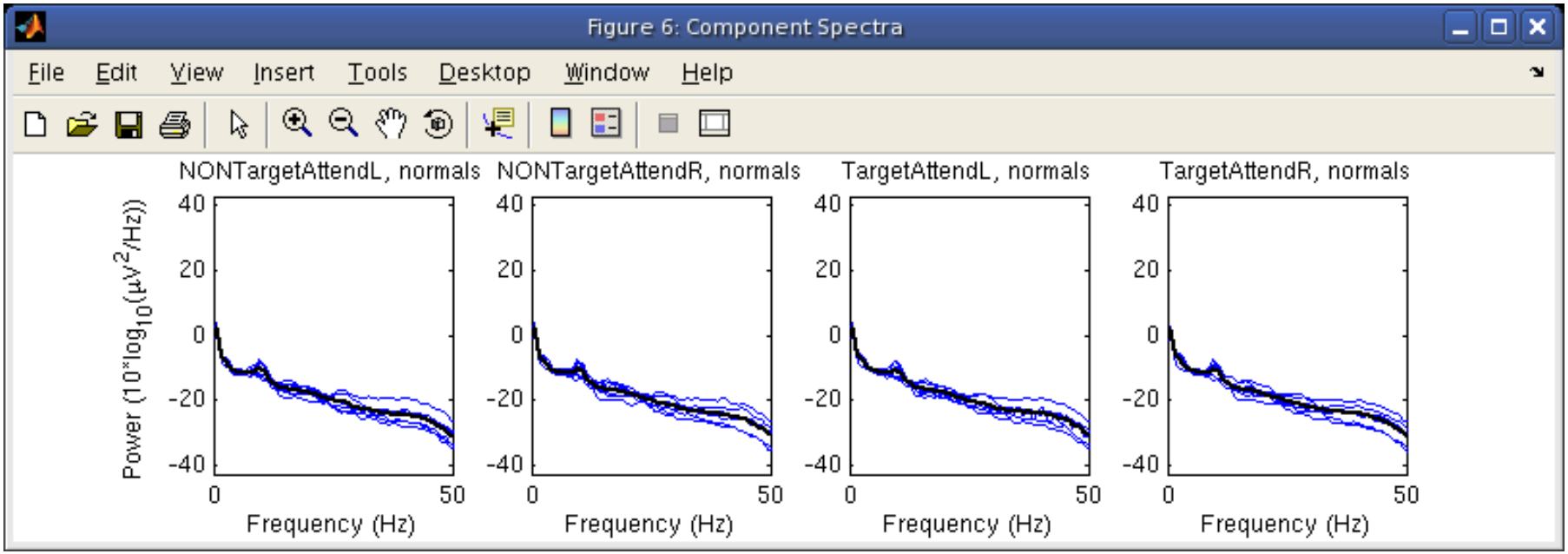
Plot cluster spectra





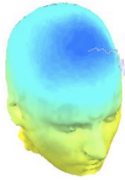
Plot cluster spectra

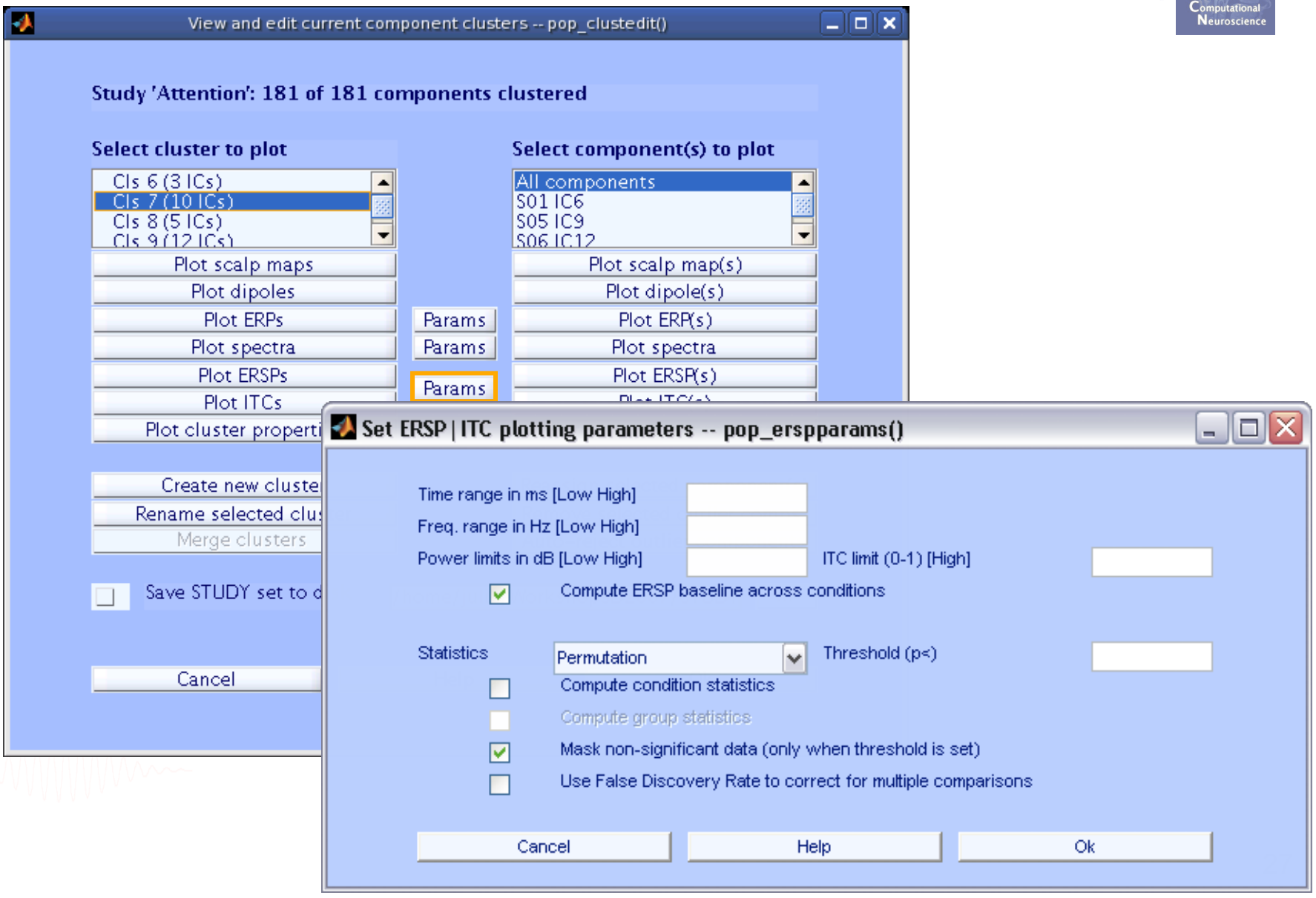




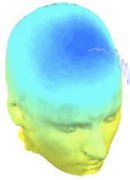
Each blue trace is the power spectrum of a different component

Plot cluster ERSPs



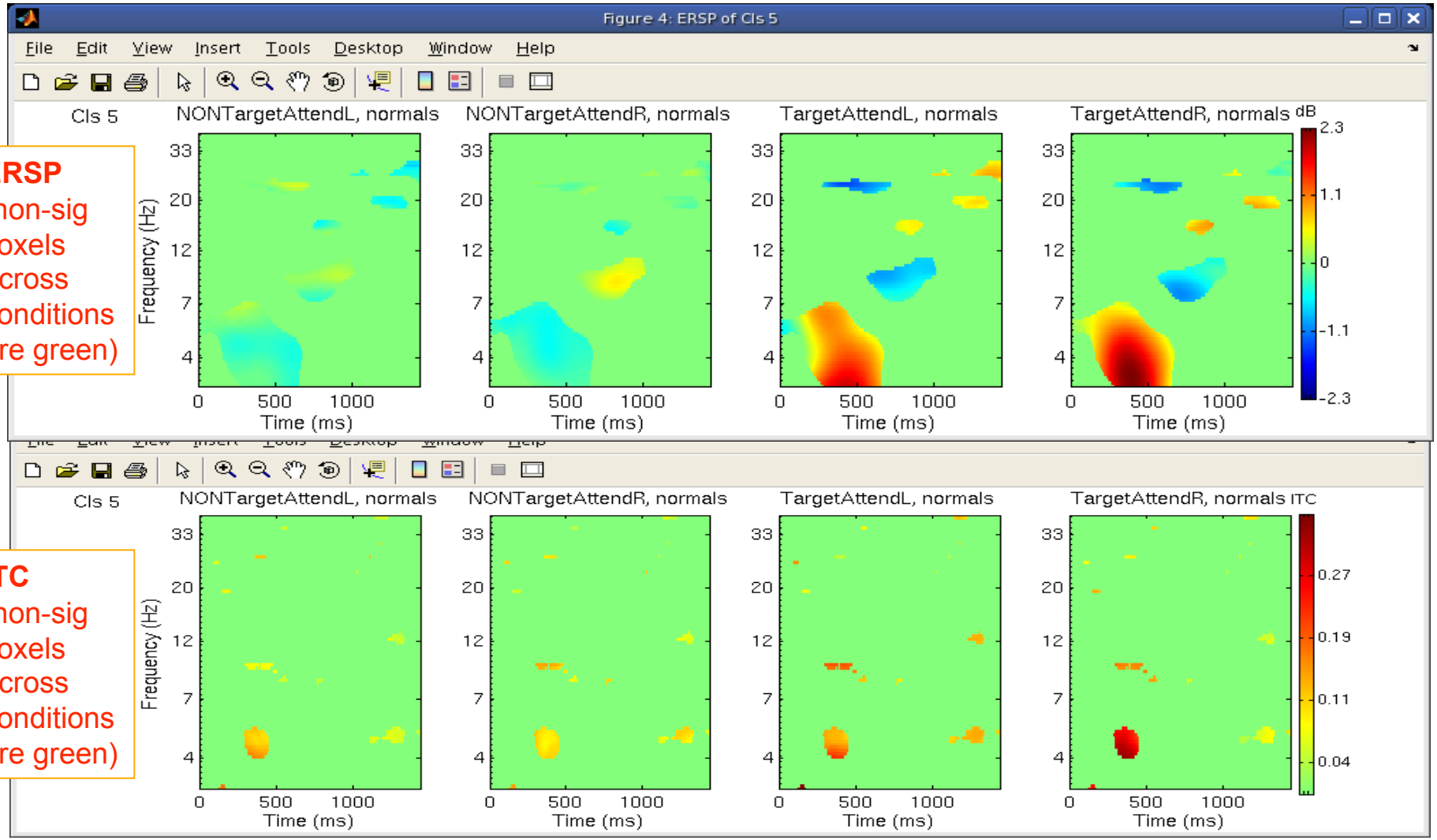


Plot cluster ERSPs

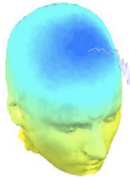


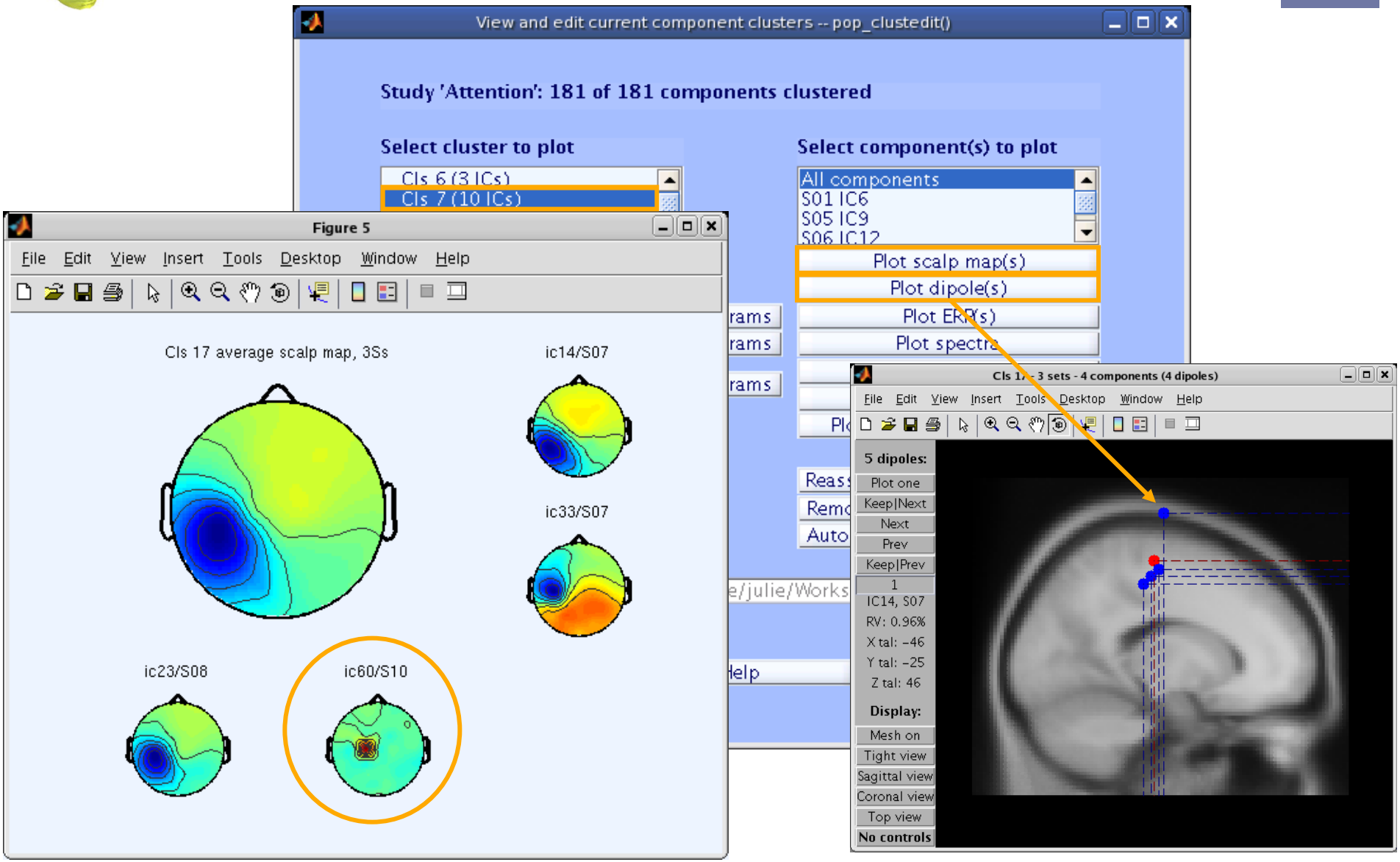
ERSP (non-sig voxels across conditions are green)

ITC (non-sig voxels across conditions are green)

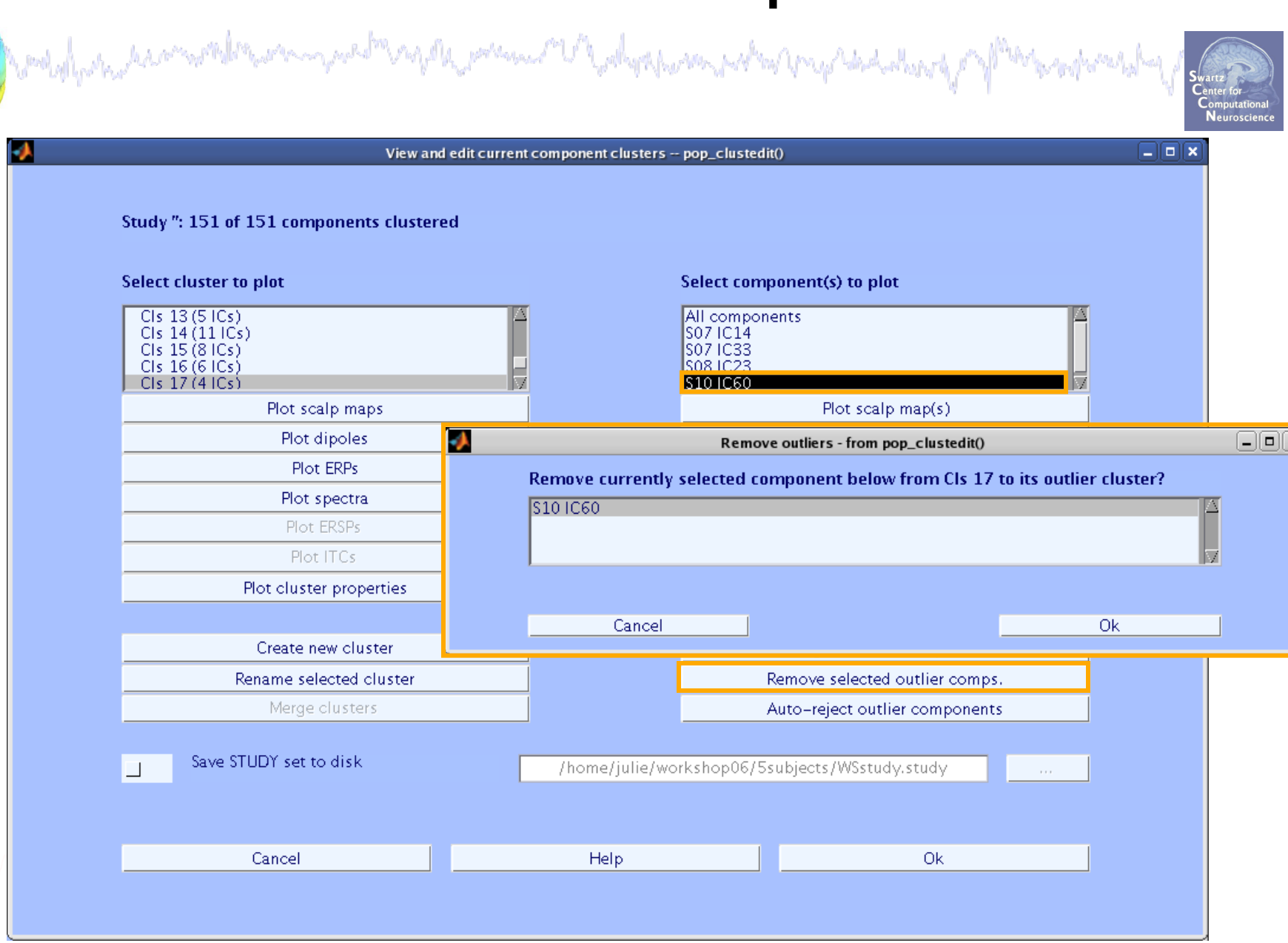


Remove outlier components

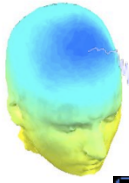


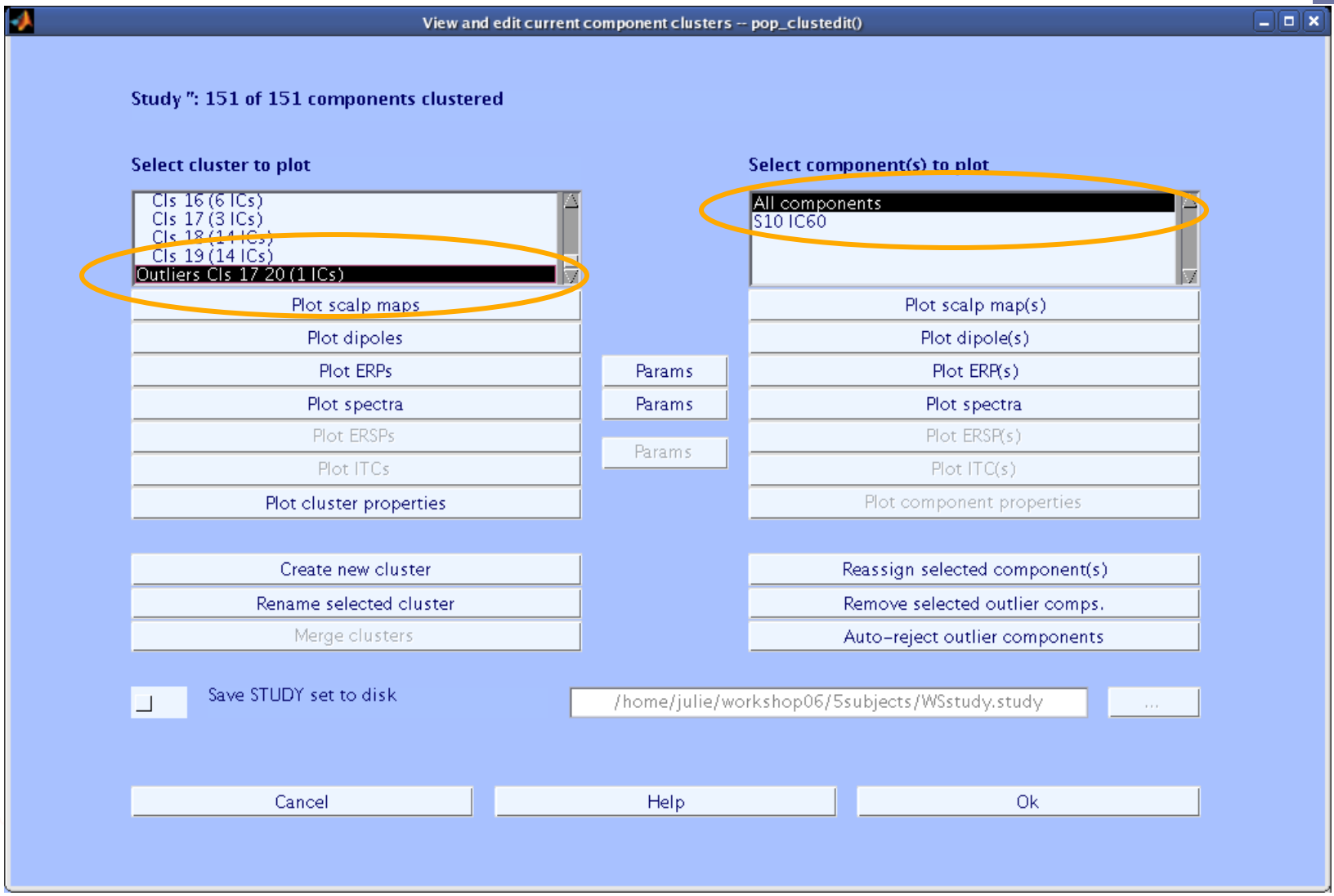


Remove outlier components

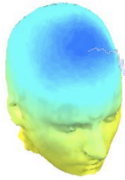


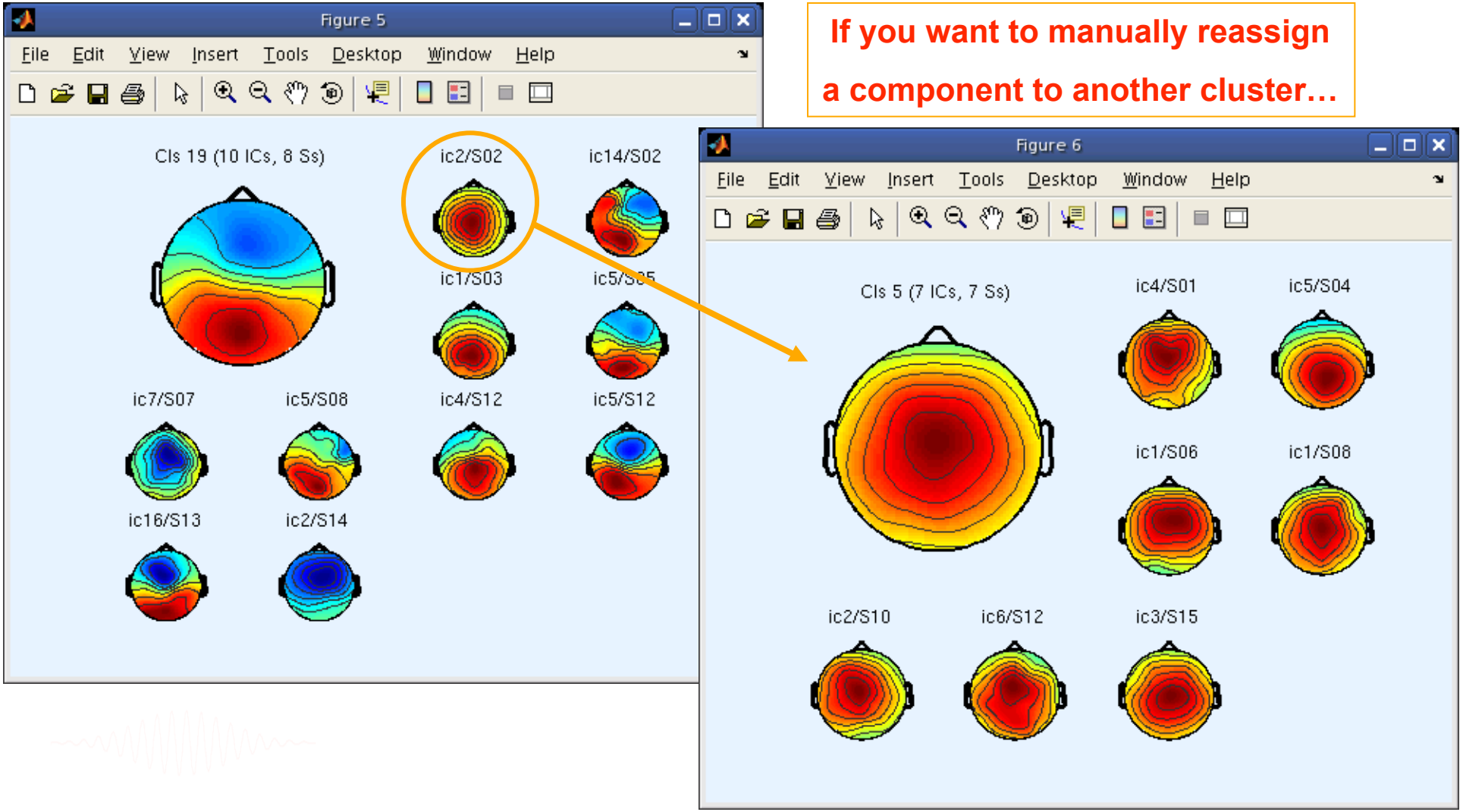
Remove outlier components





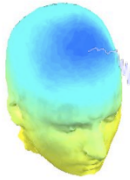
Reassign component

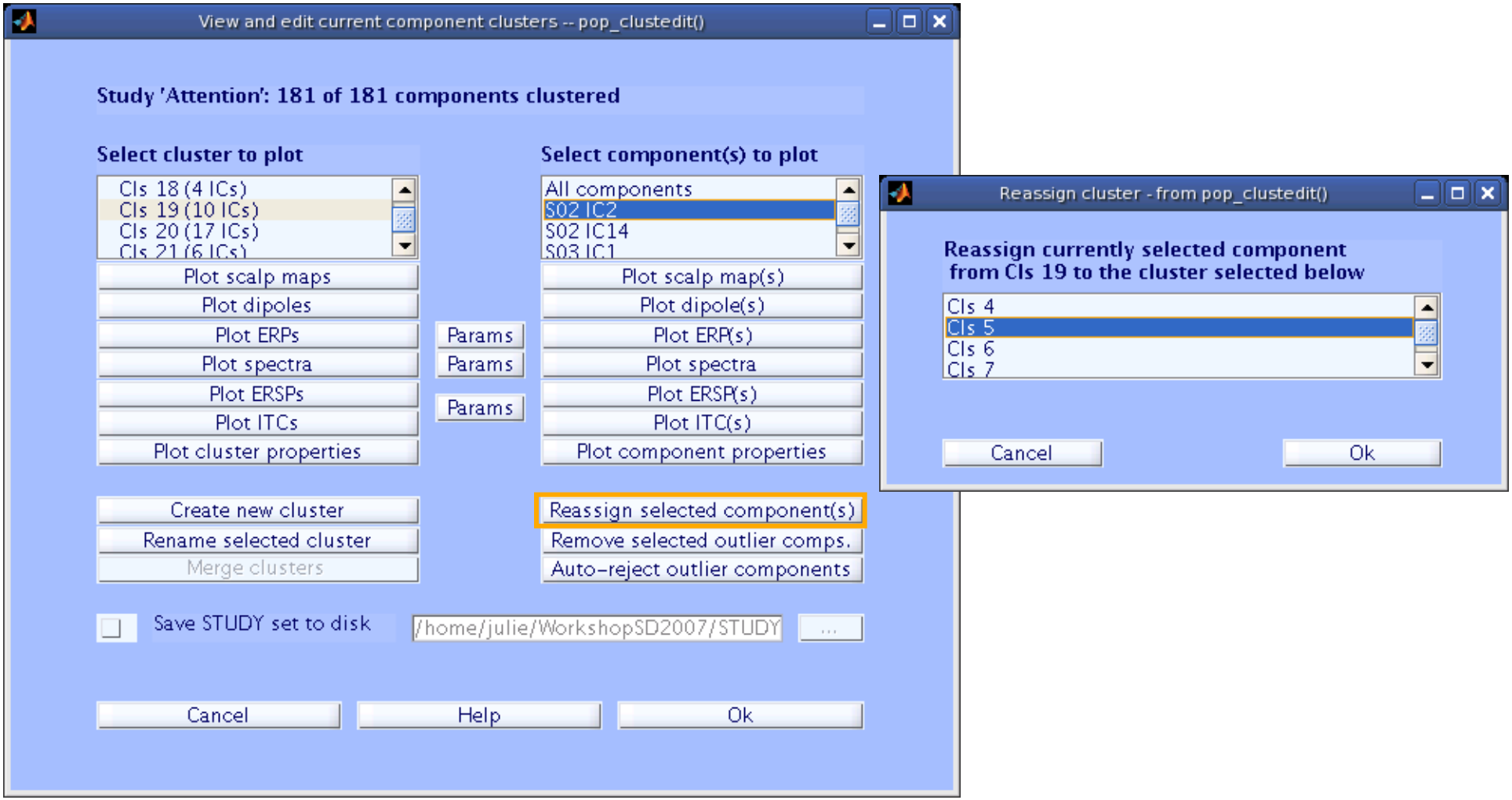




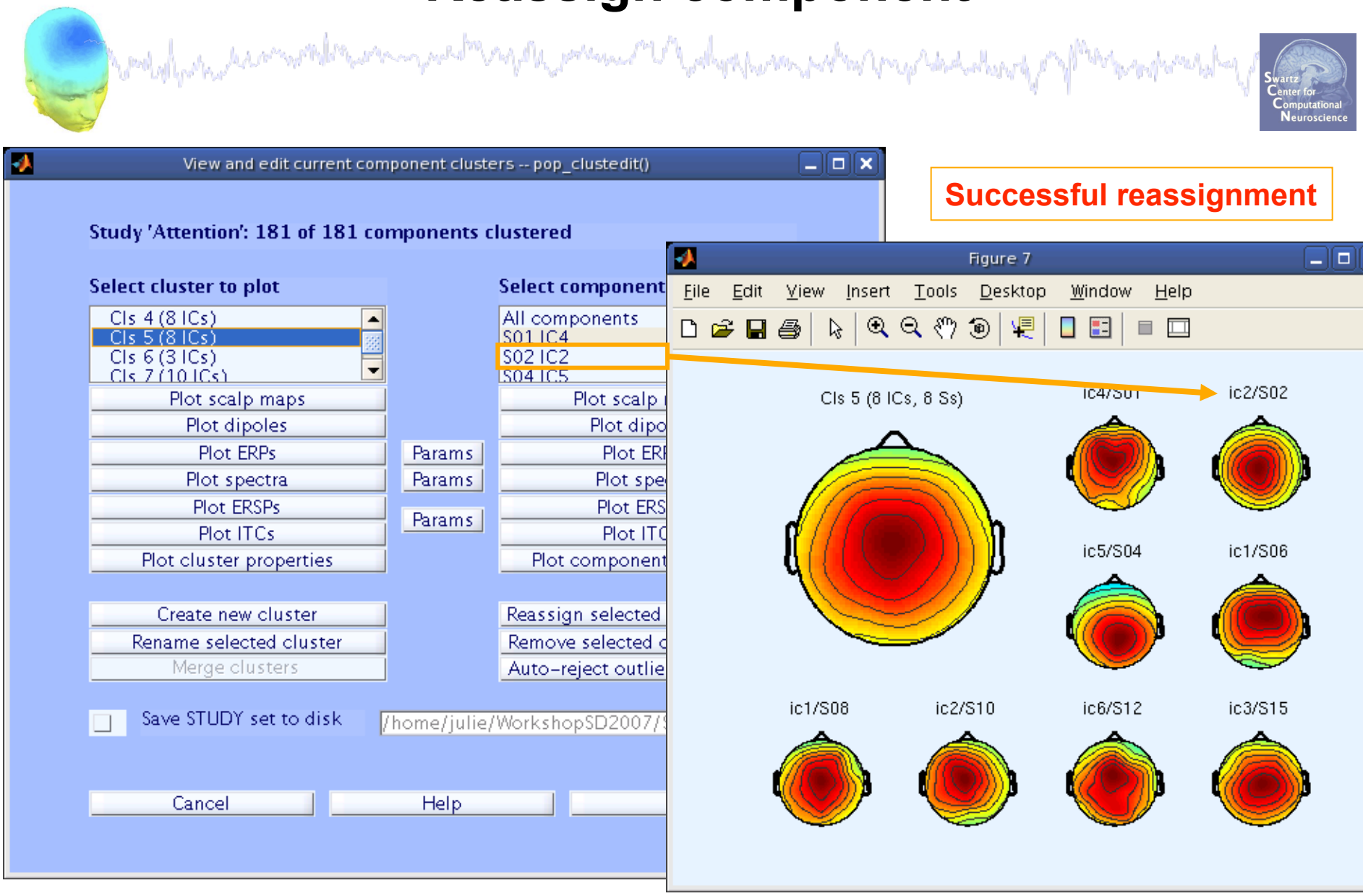
If you want to manually reassign a component to another cluster…

Reassign component

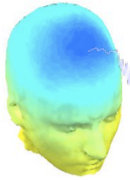


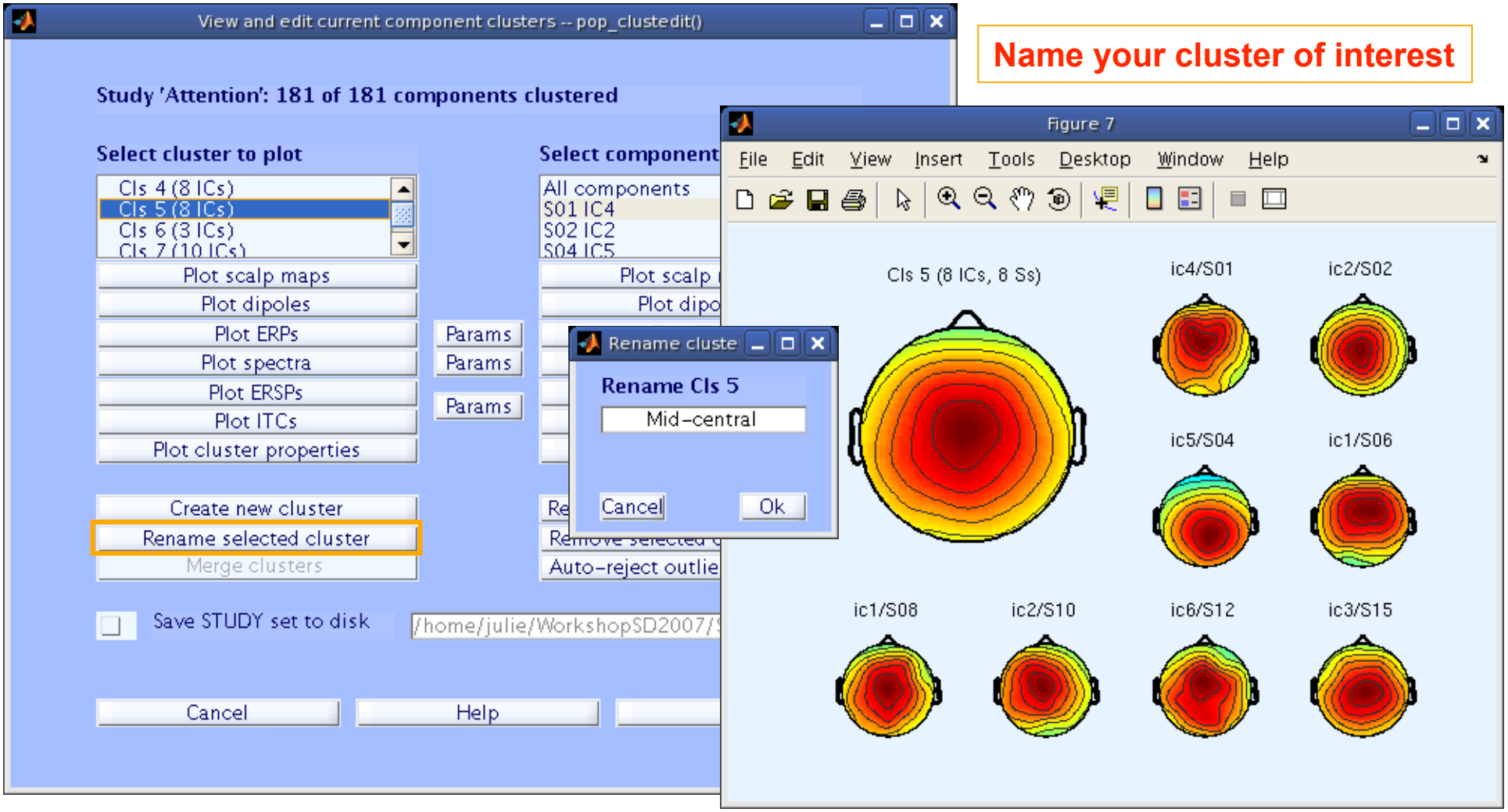


Reassign component



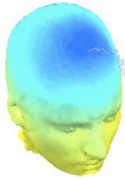
Rename a cluster



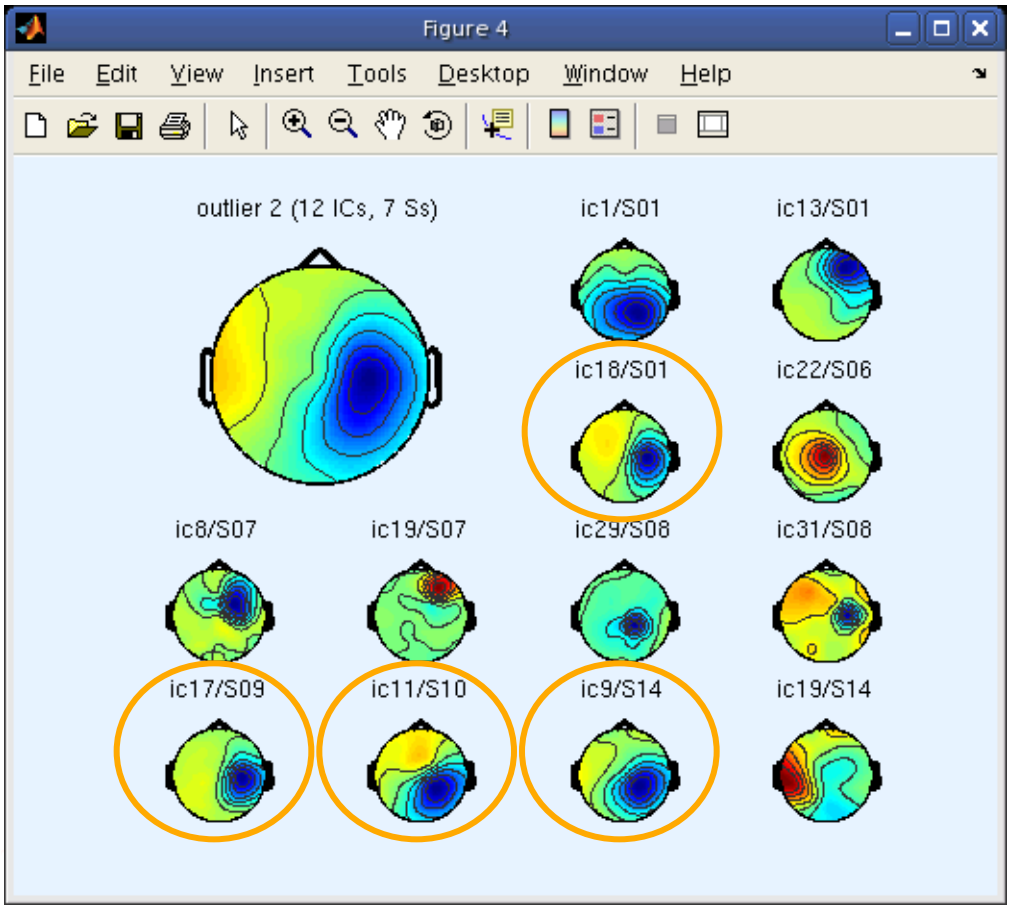


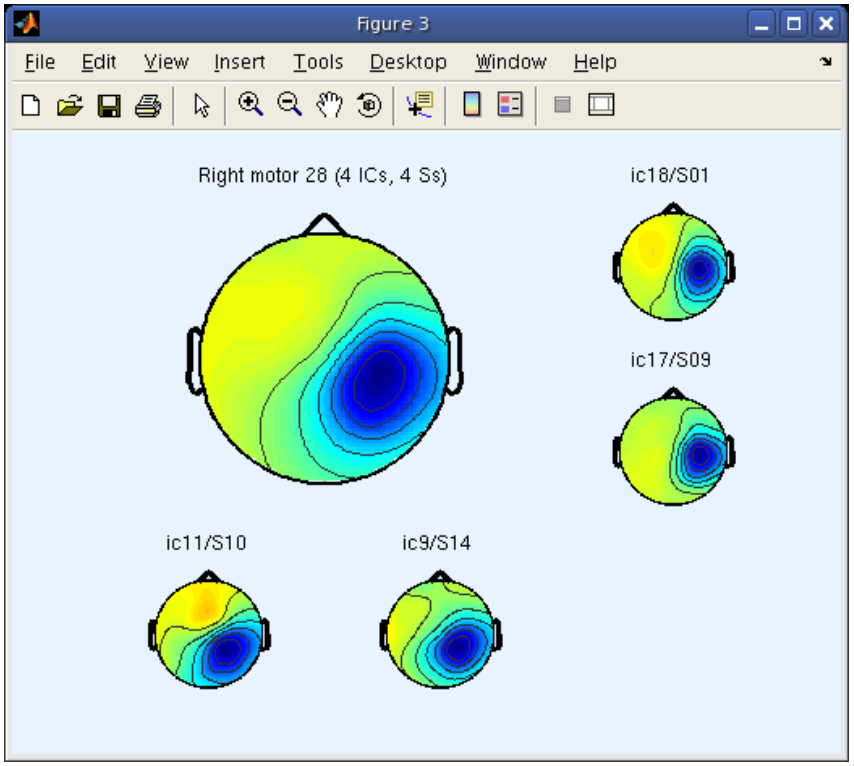
Name your cluster of interest

Create a new cluster



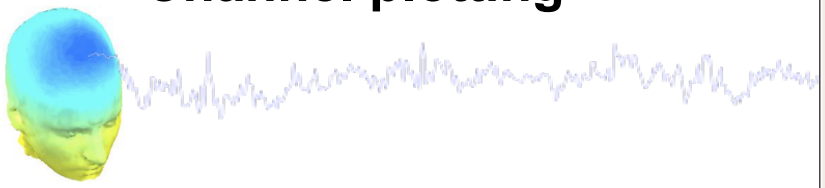
You found a bunch of 'outliers' that seem well-matched

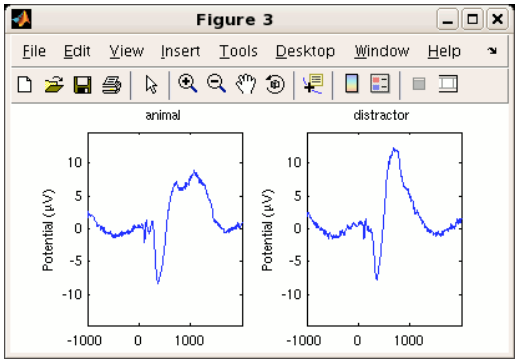


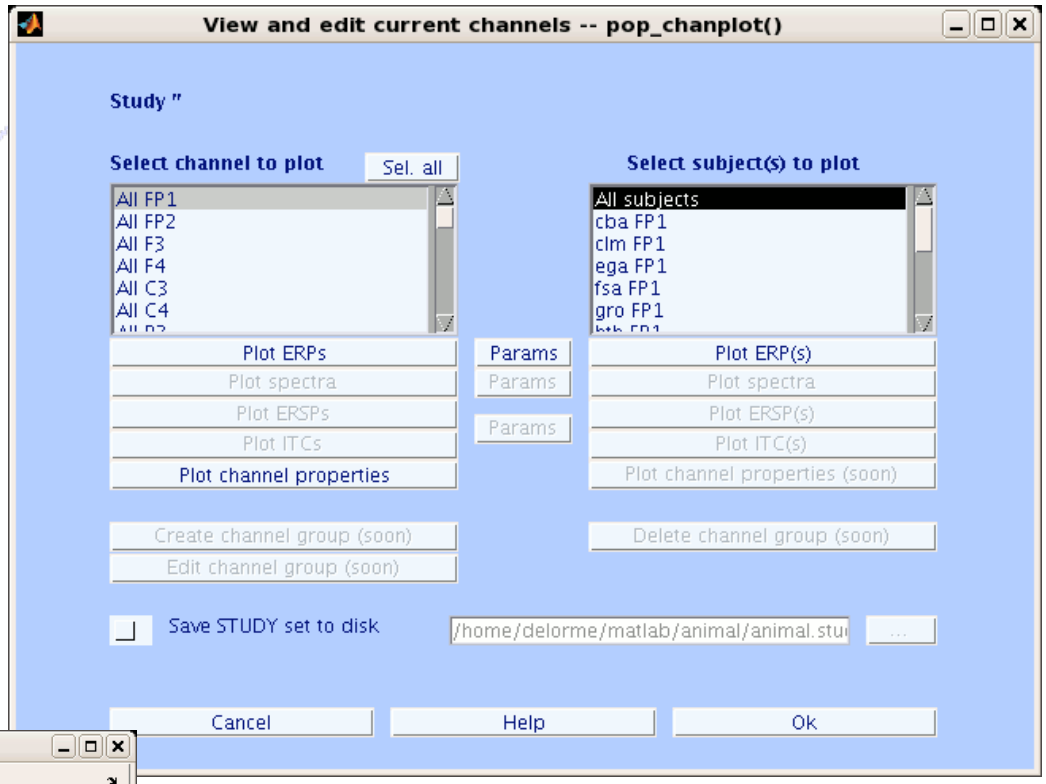


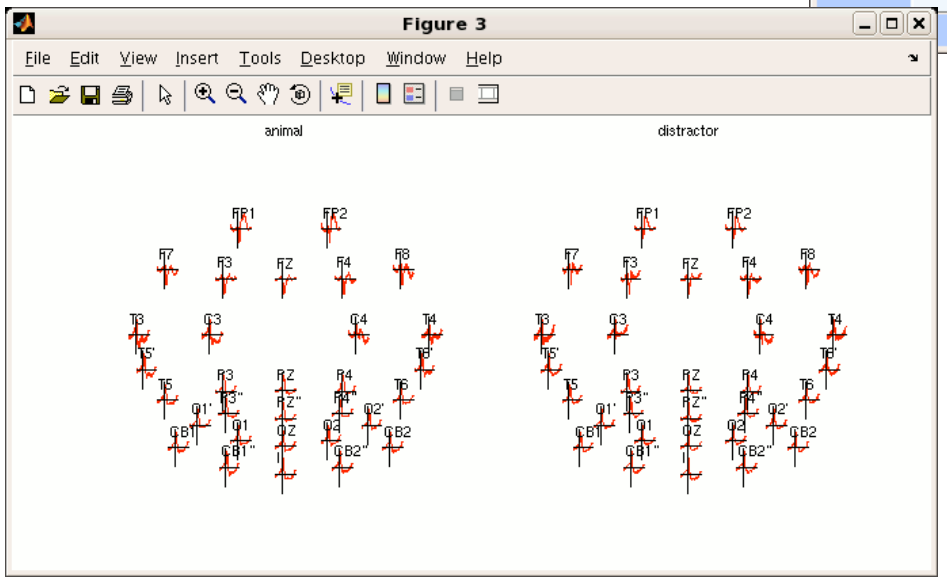
Create a new cluster…

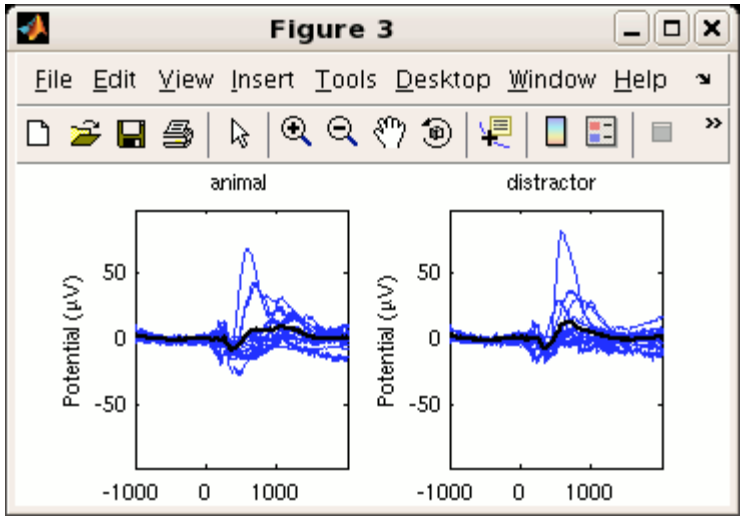
Channel plotting











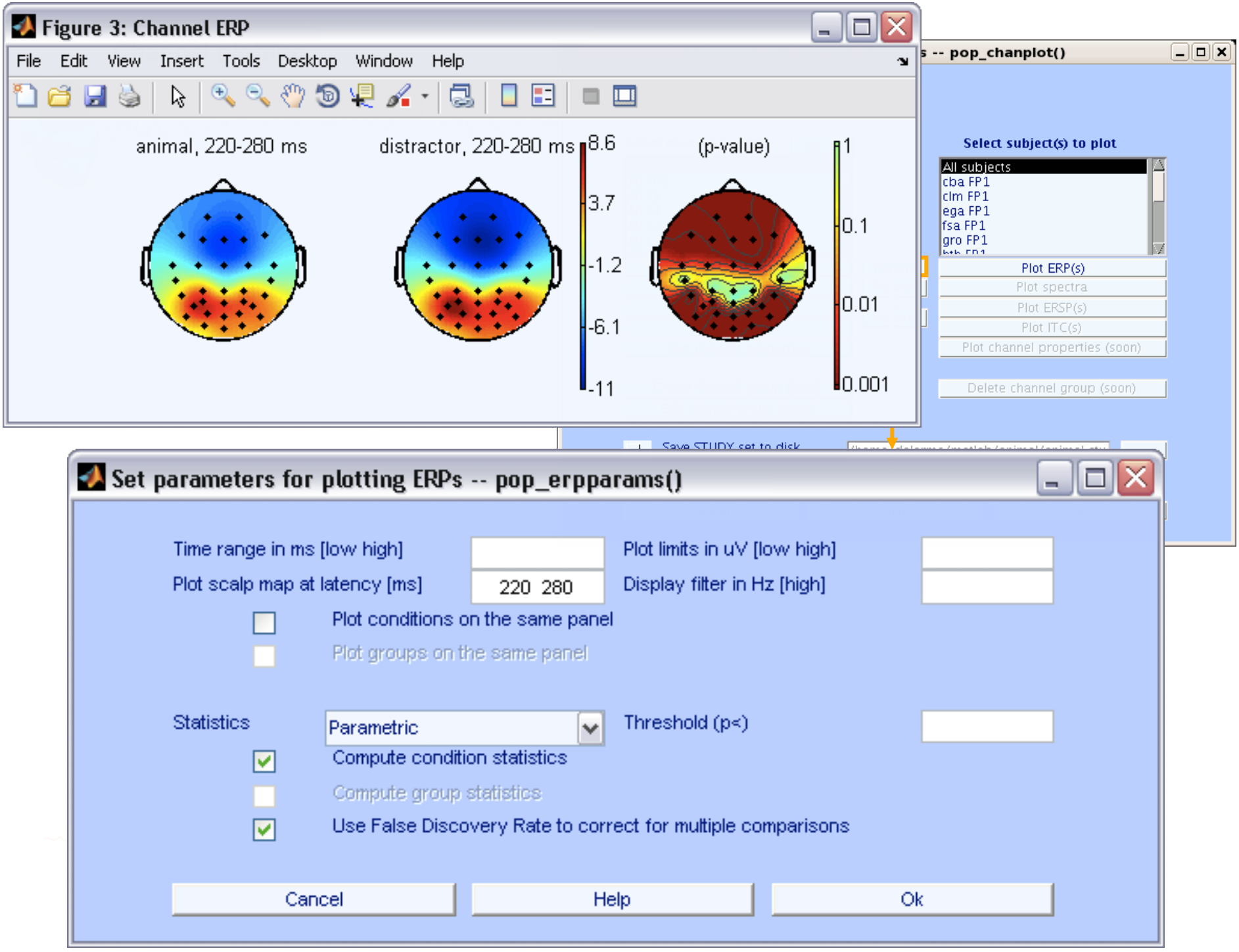
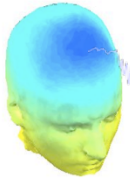


Figure 3: Channel ERP
File Edit View Insert Tools Desktop Window Help
animal, 220-280 ms
distractor, 220-280 ms
(p-value)
8.6
3.7
-1.2
-6.1
-11
1
0.1
0.01
0.001
-- pop_chanplot()
Select subject(s) to plot
All subjects
cba FP1
clm FP1
ega FP1
fsa FP1
gro FP1
Plot ERP(s)
Plot spectra
Plot ERSP(s)
Plot ITC(s)
Plot channel properties (soon)
Delete channel group (soon)
Set parameters for plotting ERPs -- pop_erpparams()
Time range in ms [low high]
Plot scalp map at latency [ms]
220 280
Plot limits in uV [low high]
Display filter in Hz [high]
Plot conditions on the same panel
Plot groups on the same panel
Statistics
Parametric
Threshold (p<)
Compute condition statistics
Compute group statistics
Use False Discovery Rate to correct for multiple comparisons
Cancel
Help
Ok

Exercise



- **Novice**

  - Use the GUI to build a STUDY (for practice, try just a few subjects).

  - Choose 'precompute' options (but do not recalculate for the sake of time).

  - Choose 'precluster' options and cluster.

  - Use the GUI to plot cluster and component data using default parameters

- **Intermediate**

  - Create a Study and add a single subject. Get the history from the command line and script a loop to build a STUDY with all subjects from the commandline

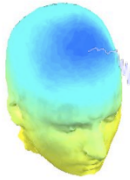
  - Precluster (pre-computation already done) and cluster components using measures of your choice.

  - Use the GUI to plot cluster and component data trying out different plotting parameters such as x/y-axis limits, and color scale limits to compare absolute values across clusters.

  - Apply statistical thresholds of your choice

  - Create and name a new cluster, fill with your choice of ICs

STUDY analysis, Part 1



Swartz Center for Computational Neuroscience

Task 1

Script STUDY plotting functions

Task 2

STUDY structure details

Task 3

Load/plot data from commandline

Task 4

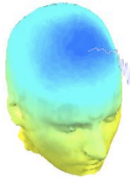
Apply significance limits

Task 5

Time/frequency analysis within cluster

Exercise...

STUDY analysis, Part 1



Task 1

Script STUDY plotting functions

Task 2

STUDY structure details

Task 3

Load/plot data from commandline

Task 4

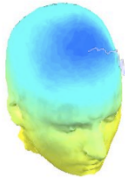
Apply significance limits

Task 5

Time/frequency analysis within cluster

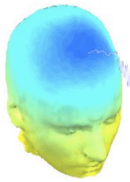
Exercise...

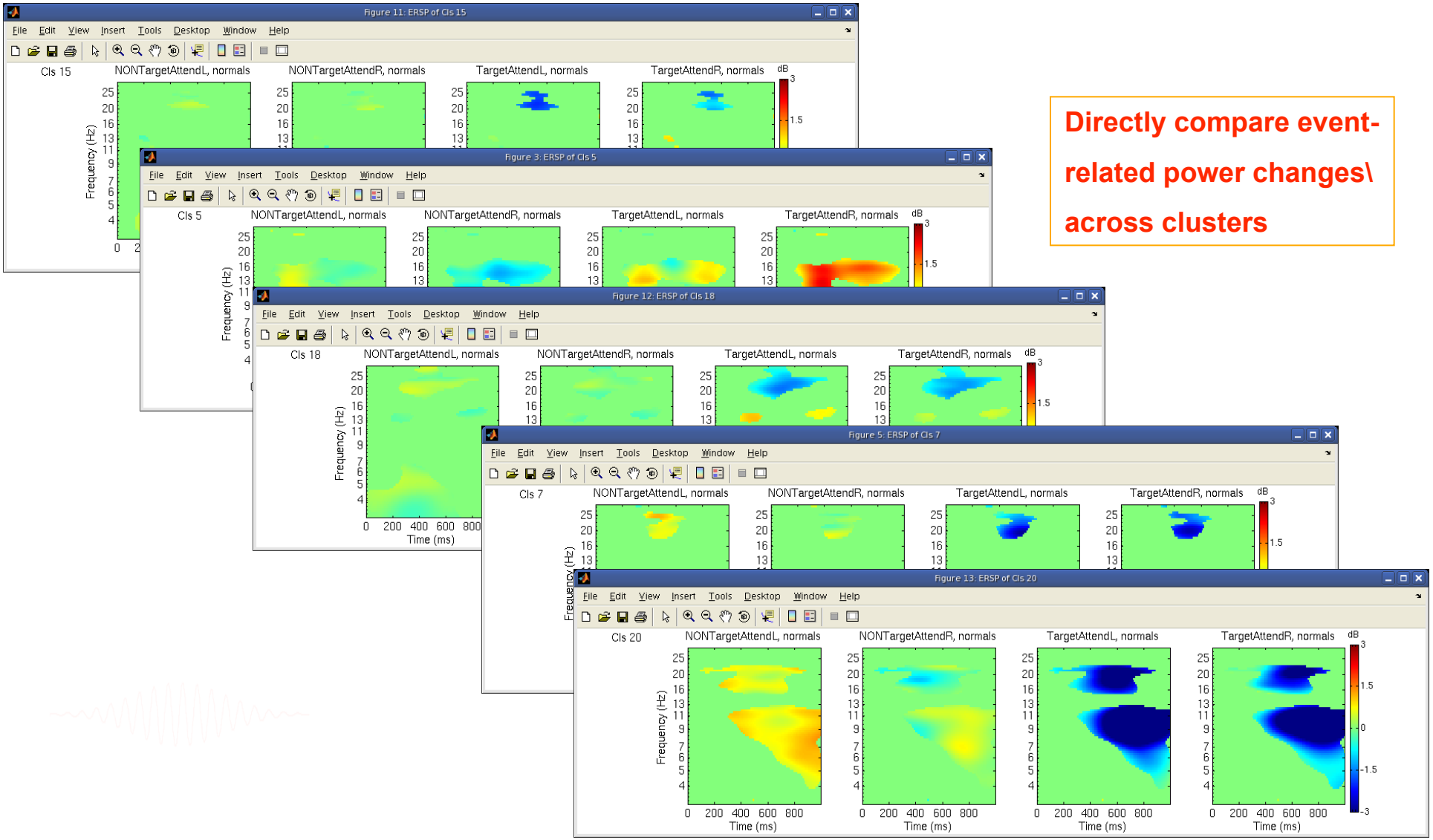
Task 1: Plot cluster ERSP



```
% plot all cluster ERSPs with variable parameters using a for
% loop to compare power across clusters:--------------------

% Define variables:-----------------------------------------
frqlim = [3 30]; % ERSP freq range (Hz)
tmlim = [0 1000]; % ERSP time range (ms)
maxdb = 4; % set color limit for dB power from baseline
%---------------------------------------------------------

for clust = 3:length(STUDY.cluster)
     STUDY = std_erspplot(STUDY,ALLEEG, 'clusters',clust, …
    'plotsubjects', 'on', 'statistics', 'perm', …
    'timerange',tmlim,'freqrange', frqlim, 'threshold',0.01, …
    'ersplim',[-maxdb maxdb]);
end;
```

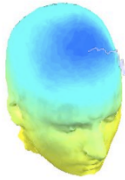
Task 1: Cluster ERSP plotting





Directly compare event-related power changes\ across clusters

STUDY analysis, Part 1



Swartz Center for Computational Neuroscience

Task 1

Script STUDY plotting functions

Task 2

STUDY structure details

Task 3

Load/plot data from commandline

Task 4

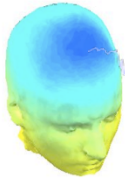
Apply significance limits

Task 5

Time/frequency analysis within cluster

Exercise...

Task 2: STUDY structure details



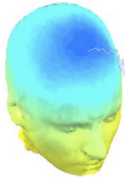
Question:

I want to know which ICs from which

subjects are in a particular cluster.

Where in the STUDY structure can I find this information?

Task 2: Understanding STUDY structure



Choose a cluster:

```
>> STUDY.cluster
ans =
```

27 total clusters

```
1x27 struct array with fields:
    name
    parent
    child
    comps
    sets
    algorithm
    centroid
    preclust
    topo
    topox
    topoy
    topoall
    topopol
```

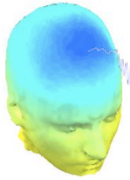
```
>> STUDY.cluster(clust)
ans =
 name: 'Cls 3'
       parent: {'ParentCluster 1'}
        child: []
        comps: [20 19 10 9 1 4]
         sets: [4x6 double]
    algorithm: {'Kmeans'  [25]}
     centroid: []
     preclust: [1x1 struct]
     selected: 1
      allinds: {4x1 cell}
      setinds: {4x1 cell}
     erspbase: {4x1 cell}
      erpdata: {4x1 cell}
     erptimes: [768x1 double]
         topo: [67x67 double]
        topox: [67x1 double]
        topoy: [67x1 double]
      topoall: {1x6 cell}
      topopol: [-1 1 -1 1 1 1]
     erspdata: {4x1 cell}
    erspfreqs: [1x126 double]
    ersptimes: [1x200 double]
```

4 = # conditions

IC indices

dataset indices

Task 2: Understanding STUDY structure



```
>> STUDY.cluster(clust)
ans =
 name: 'Cls 3'
       parent: {'ParentCluster 1'}
        child: []
        comps: [20 19 10 9 1 4]
         sets: [4x6 double]
    algorithm: {'Kmeans'  [25]}
     centroid: []
     preclust: [1x1 struct]
     selected: 1
      allinds: {4x1 cell}
      setinds: {4x1 cell}
     erspbase: {4x1 cell}
      erpdata: {4x1 cell}
     erptimes: [768x1 double]
         topo: [67x67 double]
        topox: [67x1 double]
        topoy: [67x1 double]
      topoall: {1x6 cell}
      topopol: [-1 1 -1 1 1 1]
     erspdata: {4x1 cell}
    erspfreqs: [1x126 double]
   ersptimes: [1x200 double]
```

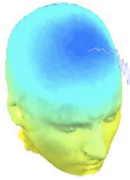
```
>> STUDY.cluster(clust).setinds{cond}

ans =

    14    38    46    50    54    54
```

Dataset 38!!

```
>> STUDY.cluster(clust).allinds{cond}

ans =

    20    19    10    9    1    4
```

```
>> STUDY.datasetinfo(38) % access dataset 38
ans =
     filepath: '…/Workshop/STUDY/S10'
     filename: 'S10_attend1_pos5.set'
      subject: 'S10'
      session: 1
    condition: 'NONTargAttnL'
        group: 'normals'
        index: 38
        comps: [1 2 3 4 5 6 7 …
```

STUDY analysis, Part 1



Task 1

Script STUDY plotting functions

Task 2

STUDY structure details

Task 3

Load/plot data from commandline

Task 4

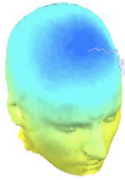
Apply significance limits

Task 5

Time/frequency analysis within cluster

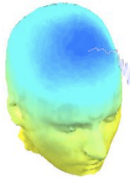
Exercise...

Task 3: Load data from commandline



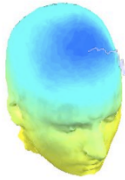
```
** Where is the raw data stored?
Data for each subject is stored in the file path
of that subject (STUDY.datasetinfo(subj).filepath)
** How to load it from the commandline:
File name format:  'setname.extension'
extension = '.ica*' or '.dat*' (for channel data)
for example:
S01_attend1_pos1.icaerp  % ERP data
S01_attend1_pos1.icaersp % ERSP data
S01_attend1_pos1.icaitc  % ITC data
S01_attend1_pos1.icaspec % Power spectrum data
S01_attend1_pos1.icatopo % Scalp map data

% Example of channel data file name:
S01_attend1_pos1.daterp  % ERP data
```

Task 3: Load individual ERSPs



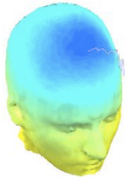
```
% call in ERSP data for all ICs in a single cluster:
clust = 5; % choose a cluster
cond = 1; % choose experimental condition
tmlims = [0 1000];  % time limits (ms)
frqlims = [0 40];   % frequency limits (Hz)

for ic = 1:size(STUDY.cluster(clust).sets,2)
   setidx = STUDY.cluster(clust).setind{cond}(ic);
   comp = STUDY.cluster(clust).allinds{cond}(ic);
   [logersp(:,:,ic), logfreqs, timevals, params, baseersp] = …
   std_readersp(ALLEEG, setidx, comp, tmlims, frqlims);
end;
```

Task 3: Load individual ERSPs



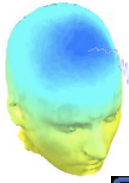
```
% Check imported variables in workspace:

>> whos logersp logfreqs timevals params baseersp

  Name            Size                    Bytes  Class

  baseersp       91x1                       728  double
  logersp        91x106x7                540176  double
  logfreqs        1x91                       728  double
  params          1x1                       4432  struct
  timevals        1x106                      848  double
```

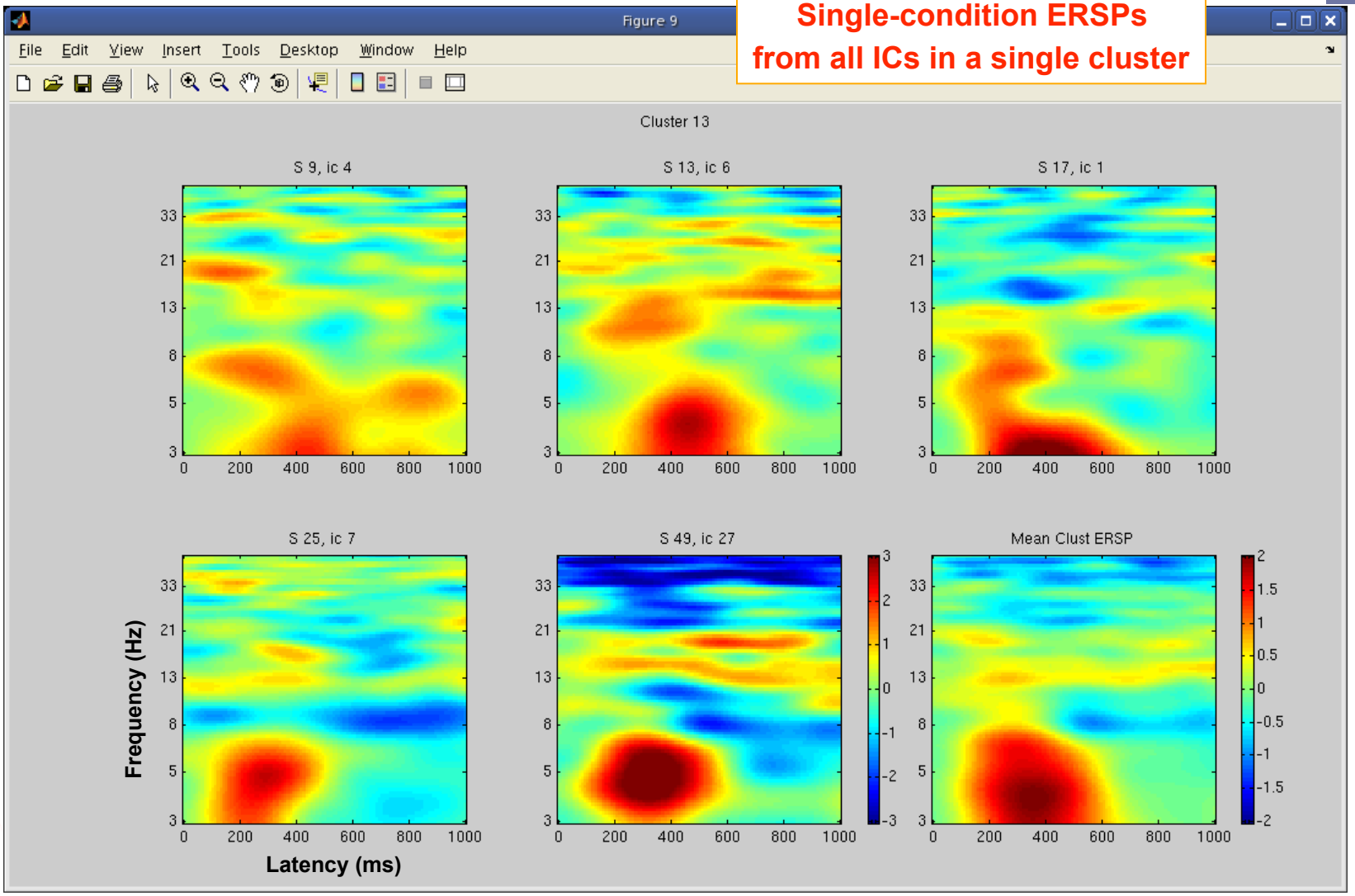
Task 3: PLOT individual ERSPs



```
% Plot the imported data:----------

figure; clim = 3; % standardize color limits

for ic = 1:size(logersp,3)
    sbplot(row,col,ic);
    imagesclogy(timevals, logfreqs, logersp(:,:,ic)); % see tftopo
    set(gca,'clim', [-clim clim]); % adjust the color limits
    set(gca,'ydir','norm'); % plot low freqs at the bottom
end;
cbar; % include a colorbar

% plot the mean for comparison:
sbplot(row,col,ic+1)
imagesclogy(timevals, logfreqs, mean(logersp,3));
```

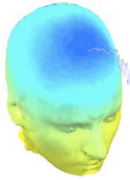
Task 3: PLOT individual ERSPs





Single-condition ERSPs from all ICs in a single cluster

STUDY analysis, Part 1



Task 1

Script STUDY plotting functions

Task 2

STUDY structure details

Task 3

Load/plot data from commandline

Task 4

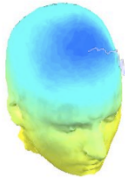
Apply significance limits

Task 5

Time/frequency analysis within cluster

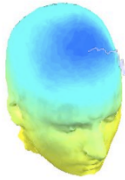
Exercise...

Task 4: Requires access to .mat files



```
% Load ERSP data as a .mat file

% Variables:
cond = 1; % which experimental condition
subj = 1; % which subject

% files are all .mat format:
load_string = [basedir,subjs{subj},'/',…
      subjs{subj},'_',setnames{cond}(1:end-4),'.icaersp'];
% actual string: '…/STUDY/S01/S01_attend1_pos1.icaersp'

ERSPdata = load('-mat',load_string);
```

Task 4: Raw data structure



```
>> ERSPdata
    comp1_ersp: [126x200 single]
 comp1_erspbase: [1x126 single]
 comp1_erspboot: [126x2 single]
    comp2_ersp: [126x200 single]
 comp2_erspbase: [1x126 single]
 comp2_erspboot: [126x2 single]
          freqs: [1x126 double]
          times: [1x200 double]
       datatype: 'ERSP'
     parameters: {1x26 cell}
       datafile: [1x57 char]
>>
```

ERSP dB data → comp1_ersp

dB baseline → comp1_erspbase

bootstrap limits → comp1_erspboot

126 frequency bins → freqs

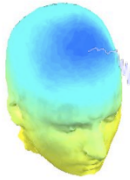
200 time points → times

comp1_ersp → 200 time points

comp1_erspboot → upper and lower bootstrap limits

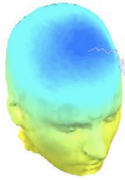
comp2_ersp → 126 frequency bins

Task 4: ERSP significance limits



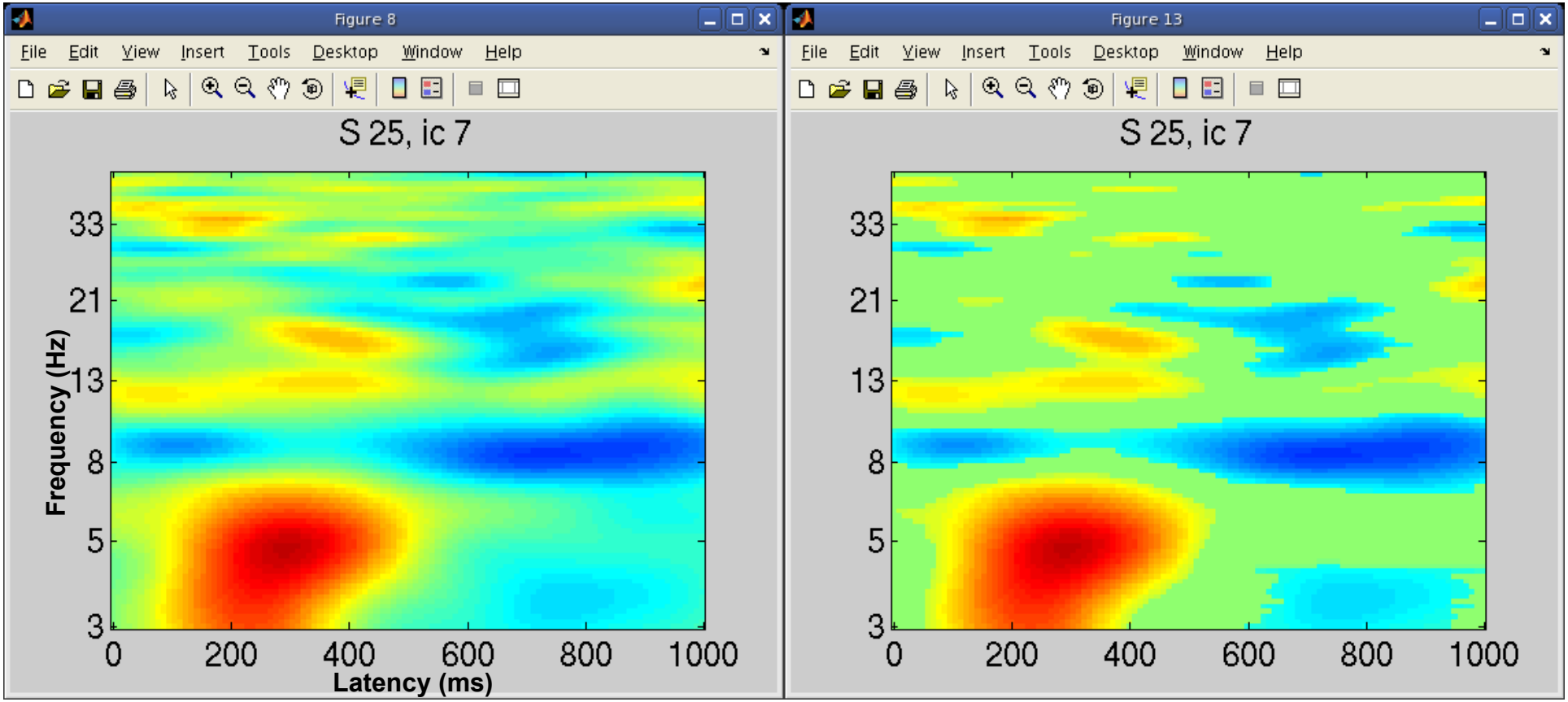
```
>> ERSPdata
comp1_ersp: [126x200 single]
comp1_erspbase: [1x126 single]
comp1_erspboot: [126x2 single] % (freqs x lower/upper limits)
%-------------------------------------------------------
% create min and max limit matrices:-----------------------

minmask  = repmat(ERSPdata.comp1_erspboot(:,1),…
         [1 size(ERSPdata.comp1_ersp,2)]);

maxmask = repmat(ERSPdata.comp1_erspboot(:,2),…
         [1 size(ERSPdata.comp1_ersp,2)]);
%-------------------------------------------------------
sig_ersp = ERSPdata.comp1_ersp; % extract relavent ERSP

% zero out values within significance limits:---------------
sig_ersp(find(sig_ersp > minmask & sig_ersp < maxmask)) = 0;

% plot the results-----------------------------------------
figure; imagesclogy(ERSPdata. times, ERSPdata. freqs, sig_ersp);
%
```

Task 4: Plot a single IC ERSP

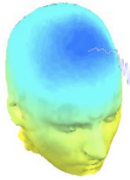


UN-masked

Masked (p = .01)



STUDY analysis, Part 1



Task 1

Script STUDY plotting functions

Task 2

STUDY structure details

Task 3

Load/plot data from commandline

Task 4

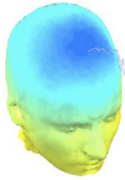
Apply significance limits

Task 5

Time/frequency analysis within cluster

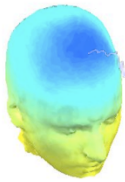
Exercise...

Task 5: Mean theta power across conditions

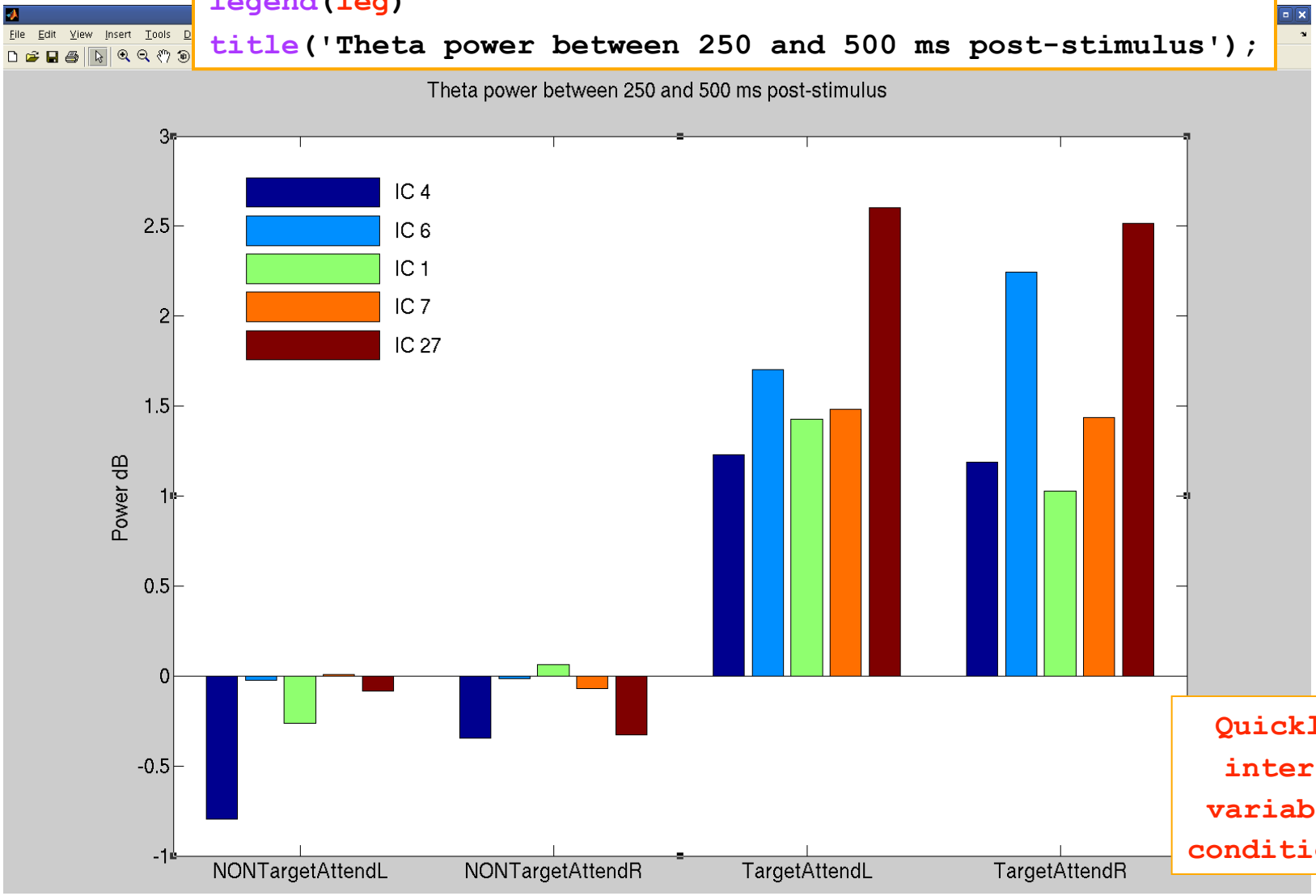


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Time/Frequency analysis:-----------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clust = 13; % cluster to analyze
tlims = [250 500]; % time limits (ms)
flims = [3 6];     % frequency limits (Hz)

for cond = 1:4
    for ic = 1:length(STUDY.cluster(clust).comps)
        setidx = STUDY.cluster(clust).setinds{cond}(ic);
        comp = STUDY.cluster(clust).comps(ic);
        [logersp(:,:,ic), logfreqs, timevals, params, baseersp] = …
         std_readersp(ALLEEG, setidx, comp, tlims, flims);
         tfdat(cond,ic) = mean(mean(logersp(:,:,ic)));
        leg{ic} = ['IC ',num2str(STUDY.cluster(clust).comps(ic))];
    end;
end;
```

Task 5: Mean theta power across conditions

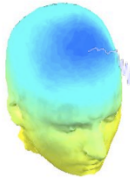


```
figure;
ph = bar([1:4],tfdat);
set(gca,'xticklabel',STUDY.condition);
legend(leg)
title('Theta power between 250 and 500 ms post-stimulus');
```



Quickly assess inter-subject variability and condition effects

Exercise



- **Novice**

  - Script a loop through clusters to plot an activity measure(s) of your choice (ie, component properties, ERSP, spectra, ERP, etc)

  - use 'eegh' to recover EEGLAB plotting function commands

- **Intermediate**

  - Plot an activity measure of your choice (ie, spectra, ERSPs, ITC, etc.) for all members of a cluster from the commandline:

    - Use *std_readersp()* or analogous STUDY function

- **Advanced**

  - Plot masked ERSPs for all members of a cluster

    - use the `load('-mat',load_string)` **command**

  - Plot mean power in a small time/frequency window across all ICs and conditions for a single cluster

** SCRIPT AVAILABLE ON THE COMMAND LINE
practicum_10.m